

**Universitat de Lleida  
Escola Politècnica Superior  
Enginyeria Tècnica en Informàtica de Gestió**

**Treball final de carrera**

**Tractament de dades i geolocalització:  
Proper bus Barcelona**

**Autor: Eduard Ereza Martínez  
Directora: Teresa Alsinet Bernadó  
Juliol del 2013**

# Índex

1. Introducció.....	4
1.1. Antecedents i context del projecte.....	4
1.1.1. Característiques de la xarxa de bus de Barcelona.....	4
1.1.2. Aplicacions existents.....	5
1.2. Objectius.....	6
1.3. Restriccions.....	6
1.4. Legalitat.....	6
2. Informació sobre les tecnologies implicades.....	8
2.1. Terminologia.....	8
2.2. Tecnologies implicades.....	9
2.2.1. Servidor de dades.....	9
2.2.2. Aplicació Android.....	10
2.3. La plataforma mòbil escollida: Android.....	10
2.3.1. Funcionament del sistema operatiu.....	10
2.3.2. Publicació d'aplicacions i seguretat.....	11
2.3.3. Components bàsics d'una aplicació.....	12
2.3.4. Entorn de desenvolupament.....	14
2.3.5. Fragmentació.....	15
3. Disseny de l'aplicació.....	16
3.1. Funcionalitat i interfície d'usuari.....	16
3.1.1. Pantalla de cerca.....	16
3.1.2. Pantalla de preferides.....	16
3.1.3. Pantalla d'una parada.....	16
3.1.4. Pantalla d'escaneig per QR.....	17
3.1.5. Pantalla del mapa.....	18
3.1.6. Pantalla de cerca manual.....	18
3.1.7. Pantalla “Quant a”.....	19
3.1.8. Disseny per a tauletes.....	19
3.2. Estructura de l'aplicació.....	19
3.2.1. Esquema general.....	20
3.2.2. Aplicació Android.....	20
3.2.2.1. Classes de l'aplicació i interacció.....	20
3.2.2.2. Estructura dels “layouts”.....	22
3.2.3. Processos d'obtenció de dades.....	24
3.2.3.1. Estructura de base de dades.....	24
3.2.3.2. Estructura de directoris.....	26
3.2.3.3. Processos.....	27
3.3. Casos d'ús i diagrames de seqüència.....	28
3.3.1. Cas d'ús “consulta d'una parada”.....	28
3.3.2. Cas d'ús “consulta des del mapa”.....	29
3.3.3. Cas d'ús “consulta per QR”.....	29
3.3.4. Cas d'ús “consulta per cerca manual”.....	30
3.3.5. Altres casos d'ús.....	30
4. Aspectes d'implementació.....	31
4.1. Implementació dels processos.....	31
4.1.1. Procés “fetch_all_data”.....	31
4.1.2. Processos “cities_list”, “companies_list” i “lines_list”.....	34

4.1.3. Procés “nearest_stop” .....	34
4.1.4. Procés “routes_list” .....	34
4.1.5. Procés “stops_list” .....	34
4.1.6. Procés “search_stop” .....	34
4.1.7. Procés “stop_info” .....	35
4.2. Implementació del refresc de les dades des de l'aplicació .....	36
4.3. Implementació de la interfície en mode mapa .....	36
4.4. Implementació de l'escaneig per QR .....	36
4.4.1. Obertura de QR interna .....	36
4.4.2. Obertura de QR externa .....	37
4.4.3. Extracció del codi des de les URLs .....	37
4.5. Implementació de la coloració de les línies .....	38
4.6. Dificultats trobades en la implementació .....	39
4.6.1. Canvis en la web de l'AMB .....	40
4.6.2. “Bugs” de les biblioteques .....	40
4.6.3. Problema amb la publicitat en un dispositiu .....	42
4.7. Límits de la implementació .....	42
4.7.1. Durada de l'execució i límits d'ample de banda .....	42
4.7.2. Direcció de les línies de bus .....	42
4.8. Possibles millores .....	42
4.8.1. Evitar múltiples crides al servei de l'ICC .....	43
4.8.2. Comentaris al codi .....	43
4.8.3. Ús de “caché” al mapa .....	43
5. Temporalització i cost de desenvolupament .....	44
5.1. Cost de desenvolupament .....	44
5.2. Monetització: opcions i elecció .....	45
5.3. Cost d'actualització .....	45
6. Conclusions i possibles extensions .....	46
6.1. Conclusions .....	46
6.2. Extensions futures .....	46
Apèndix 1. Instal·lació del servidor de dades .....	48
Apèndix 2. Instal·lació de l'entorn de desenvolupament .....	50
Bibliografia .....	52

# 1. Introducció

Sovint les coses més quotidianes, com ara agafar l'autobús, es poden enfocar de manera diferent, per tal de facilitar que els usuaris tinguin una informació més precisa i interessant. Per exemple, sabent quant de temps falta perquè passi el nostre autobús, podem planificar millor el nostre desplaçament i aprofitar el temps restant per fer altres coses.

Aquest és el principal objectiu d'aquest projecte: construir una aplicació que permeti conèixer amb antelació quan arribarà el proper bus a la parada on l'esperarem. Prendrem com a exemple la ciutat de Barcelona, ja que l'especial complexitat de la seva xarxa fa més necessària la construcció d'una aplicació d'aquestes característiques.

Volem construir una aplicació que resolgui les necessitats més bàsiques en aquest aspecte, i que sigui ràpida i senzilla d'utilitzar.

## 1.1. Antecedents i context del projecte

En l'actualitat, existeixen diverses aplicacions amb la finalitat d'informar a l'usuari de quant de temps disposa fins que arribi l'autobús, tant per a Barcelona com per a moltes ciutats més arreu del món. No obstant, hem cregut necessària una nova aplicació per dos motius principals: unificar criteris, ja que actualment segons quina sigui l'empresa operadora de la línia cal utilitzar una aplicació diferent; i proporcionar funcionalitats que les aplicacions oficials no permeten.

### 1.1.1. Característiques de la xarxa de bus de Barcelona

Hi ha una sèrie de característiques comunes a tota la xarxa de bus de Barcelona, independentment de qui sigui el propietari de la parada o l'empresa que opera la línia. Les podem resumir en:

- Totes les parades tenen un codi de parada independent. Es tracta d'un codi numèric d'entre una i sis xifres, que identifica la parada de manera inequívoca.
- Totes les línies de bus tenen un nom diferent. És a dir, no hi ha línies de diferents operadors amb un mateix número o nom.
- La gran majoria d'autobusos disposen de seguiment mitjançant GPS de manera que les empreses operadores són capaces de fer estimacions de quan arribarà el bus a una parada concreta.
- La majoria de parades disposen d'un codi QR amb l'identificador de la parada, tot i que depenent de qui sigui el propietari de la parada, porten a llocs web diferents.
- Algunes parades disposen de pantalles informatives amb el temps restant perquè arribi l'autobús.

Les empreses que gestionen la informació de les línies i parades són les següents:

- TMB (Transports Metropolitans de Barcelona) és l'empresa pública que gestiona i opera la majoria de parades i línies a Barcelona ciutat. És senzilla d'identificar ja que el seu color corporatiu és el vermell.
- L'AMB (Àrea Metropolitana de Barcelona) és qui gestiona la resta de parades i línies de l'àrea metropolitana. Està integrada pel conjunt d'empreses operadores, així com per diversos ens públics. El seu color corporatiu és el groc.

- Les empreses operadores, a part de TMB, són: Autobuses Horta, Mohn, Oliveras, Rosanbus, SGMT, Soler i Sauret, TCC i TUSGSAL.

### 1.1.2. Aplicacions existents

Les aplicacions existents que permeten una funcionalitat similar a la que pretenem aconseguir són les següents:

- **TMB Virtual**[1]: Aplicació oficial creada per TMB que permet consultar únicament les parades i línies de TMB. Integra tant bus com metro i permet cercar recorreguts, informació de les línies, etc. La informació de properes arribades es mostra consultant per codi de parada o per nom.
  - Punts positius: aplicació molt completa amb cerca de recorreguts i integració de correspondències entre bus i metro.
  - Punts negatius: la interfície és molt similar a una aplicació iPhone, les properes arribades sovint no es carreguen correctament o triguen a aparèixer, només mostra les parades i línies de TMB, no es poden desar parades ni línies preferides.
- **Urban Step - Barcelona**[2]: Aplicació no oficial creada per ASCII 164, que permet obtenir les dades de properes arribades dels busos.
  - Punts positius: mostra un mapa, permet desar una llista de parades preferides
  - Punts negatius: mostra totes les parades però només mostra informació de les línies de TMB, la interfície està desfasada (estil Android 2.x).
- **Buses Barcelona**[3]: Aplicació no oficial creada per Miguel Fermín que mostra les dades de properes arribades d'una manera visual.
  - Punts positius: mostra un mapa, permet desar una llista de parades preferides, mostra la informació de manera visual.
  - Punts negatius: problemes d'estabilitat, només mostra les parades i línies de TMB.
- **Buzz Barcelona**[4]: Aplicació no oficial creada per Treenamics que mostra les properes arribades de cada parada.
  - Punts positius: mostra un mapa, permet desar una llista de parades preferides, interfície actual i polida.
  - Punts negatius: només mostra les parades i línies de TMB.
- **AMB Temps bus**[5]: Aplicació oficial de l'AMB que permet obtenir les dades de properes arribades en totes les parades de Barcelona i rodalies.
  - Punts positius: unifica la informació de tots els operadors, mostra parades i línies de TMB i de la resta d'operadors, permet definir parades preferides (tot i que quan es va iniciar aquest projecte no ho permetia).
  - Punts negatius: només permet la cerca per número de parada, la interfície no és nativa.

D'altra banda, l'AMB disposa a la seva pàgina web[6] de molta més informació de línies, recorreguts, horaris, etc. que, degudament tractada, pot ser útil per proporcionar funcionalitats

addicionals en les aplicacions.

## 1.2. Objectius

L'objectiu del projecte és construir una aplicació per a Android que permeti consultar el temps restant perquè arribin les diferents línies de bus a cada parada de bus, tenint en compte les següents consideracions:

- Que sigui molt ràpid consultar la informació, introduint el mínim de paràmetres necessaris per obtenir-la.
- Que funcioni tant a Barcelona capital com a la resta de l'Àrea Metropolitana, incloent tant les parades com les línies de bus, independentment de quin sigui el seu operador.
- Que faciliti la cerca d'una parada amb antelació, sense necessitat de ser físicament a la parada, mitjançant un mapa geolocalitzat o mitjançant una cerca de text.
- Que recordi les parades preferides de l'usuari de manera que sigui encara més ràpid consultar-les per segona vegada.
- Que proporcioni el màxim d'informació que pugui ser útil a l'usuari, però sense augmentar la dificultat de les consultes més senzilles.

En resum, es tracta d'unificar en una sola aplicació la funcionalitat més útil, sovint ja oferida per alguna de les aplicacions existents, però de manera que sigui extensible a tots els operadors de transport.

## 1.3. Restriccions

Les principals restriccions del projecte vindran donades per la dificultat d'obtenció de les dades necessàries per implementar la funcionalitat.

Actualment, és possible extreure molta informació de la web de l'AMB mitjançant tècniques de *web scraping*. Així mateix, és possible extreure la informació de les línies de bus en temps real des del propi servei ofert per l'AMB a l'aplicació web AMBTempsbus[7]. En aquest projecte ens basarem en aquestes dades, tractant-les i organitzant-les com ens convingui.

No obstant, al tractar-se d'un ús no previst, és possible que en algun moment, l'AMB decideixi deixar d'oferir algunes de les dades, o bé en canviï el format sense previ avís. Aquests canvis tindrien uns resultats en l'aplicació que caldria determinar si es donés el cas.

## 1.4. Legalitat

Ja que tenim pensat fer ús de dades alienes, cal tenir en compte les possibles implicacions legals que això pot suposar. Tractant-se de serveis públics, i sempre que no fem ús del seu nom propi, generalment no sol haver-hi problemes. En aquest cas, però, n'hem volgut fer una anàlisi amb més profunditat.

A l'avís legal del lloc web de l'AMB[8], que és d'on extraïem totes les dades, podem trobar-hi la següent nota:

*“La persona usuària pot visualitzar tots els elements, imprimir-los, reproduir-los, copiar-los i emmagatzemar-los en el disc dur del seu ordinador o en qualsevol altre suport físic sempre que es respecti el context original i no es modifiquin o alterin.”*

Entenem per tant, que com a usuaris del servei, podem visualitzar, reproduir, copiar i emmagatzemar les dades, per exemple, en una base de dades, sempre que no les modifiquem. En els processos d'obtenció de dades fem alguns canvis d'estil (canviar noms de majúscules a minúscules, o eliminar els espais dintre de les cometes), però no creiem que sigui una modificació pròpiament dita ja que les dades segueixen tenint el mateix significat.

Per tant, creiem que l'ús que en fem és coherent amb les condicions proposades per l'empresa propietària de les dades.

## 2. Informació sobre les tecnologies implicades

L'aplicació proposada s'estructurarà en dues parts: l'aplicació Android en si, i un conjunt de *scripts* en un servidor que seran els encarregats d'obtenir les dades. Considerem apropiat descriure a continuació les tecnologies implicades, així com la terminologia emprada.

### 2.1. Terminologia

A continuació exposem els diferents termes emprats en aquesta memòria que poden resultar d'interès per comprendre el treball realitzat:

- **APK:** *Android Package*. Fitxer que conté una aplicació Android instal·lable.
- **Biblioteca:** Anomenada també erròniament “llibreria”, és una peça de programari que permet la seva reutilització com a component d'una altra aplicació.
- **Bug:** Error de programari no intencionat.
- **Caché o memòria cau:** Memòria o fitxers utilitzats per emmagatzemar-hi dades de consulta freqüent i que no cal que siguin en temps real, per tal d'evitar consultes innecessàries a la font de les dades. Sol tenir una data de caducitat.
- **Codi QR:** imatge codificada en forma de quadrat que pot contenir dades diverses, sovint una URL.
- **CURL:** Eina de línia d'ordres que permet obtenir el contingut d'una URL.
- **DAO:** *Data Access Object*, capa utilitzada en metodologia del programari per abstraure l'accés a la base de dades (o una altra font de dades) de la resta de l'aplicació.
- **DOM:** *Document Object Model*, és el model intern que utilitzen els navegadors per navegar per l'estructura d'una pàgina HTML.
- **Escalabilitat:** capacitat d'un servidor o servei per absorbir una forta càrrega de peticions.
- **Geolocalització:** funcionalitat que permet obtenir la ubicació de l'usuari mitjançant GPS, triangulació de xarxes mòbils o Wi-Fi.
- **JSON:** *JavaScript Object Notation*: format estàndard per a l'intercanvi de dades.
- **Latitud i longitud:** sistema de coordenades. És el sistema utilitzat per Google Maps.
- **Paràmetre GET:** tipus de paràmetre d'una URL que es passa com a part de la URL (per exemple, afegint-hi: `?camp1=valor1`)
- **Paràmetre POST:** tipus de paràmetre d'una URL que no es passa com a part de la URL sinó en les capçaleres de la petició HTTP.
- **Script:** conjunt d'ordres que permet realitzar una tasca de manera automatitzada.
- **SDK:** *Software Development Kit*. Conjunt d'elements necessaris (biblioteques, eines de compilació, etc.) per desenvolupar per a una plataforma.
- **Spinner:** component d'Android amb funcionalitat similar als “combos” tradicionals. Permet escollir un element d'una llista predefinida.



- **URL:** *Uniform Resource Locator*. Text amb un format concret que identifica un protocol d'accés, un host, port i una ruta a seguir per obtenir un element. Per exemple: `http://www.ambmobilitat.cat/AMBtempsbus/`
- **UTM ED50:** sistema de coordenades utilitzat a l'Europa occidental. És el format proporcionat per l'AMB en les seves dades de parades.
- **Web scraping:** tècnica de software consistent en extreure informació de llocs web, sovint simulant la navegació humana pel lloc web.

## 2.2. Tecnologies implicades

L'aplicació es dividirà en dos grans mòduls: el servidor de dades i l'aplicació Android. Al punt 3.2 trobareu una explicació més profunda dels diferents components, però hem cregut necessari fer una breu introducció a les tecnologies implicades en cada mòdul.

### 2.2.1. Servidor de dades

El servidor de dades serà l'encarregat d'obtenir les dades de les diferents fonts d'informació i respondre a les peticions de l'aplicació Android.

En aquest cas, hem decidit implementar un servidor GNU/Linux amb un servidor web Apache i una base de dades MySQL. El llenguatge de programació de les aplicacions necessàries serà principalment PHP, tot i que també ens ajudarem d'algun *shell script* i altres eines del sistema. A continuació descrivim les versions instal·lades i una breu descripció de cada tecnologia:

- **Sistema operatiu Ubuntu Linux 12.04:** aquesta distribució de GNU/Linux proporciona una manera senzilla i ordenada de gestionar els paquets de programari. Hem escollit l'última versió LTS (Long Term Support) per la seva estabilitat i major temps de manteniment, cosa que la fa més apropiada per a servidors.
- **Servidor web Apache 2.2:** es tracta del servidor web de codi obert per excel·lència i amb el qual és més senzilla la integració amb PHP. A part, facilita l'escalabilitat.
- **Base de dades MySQL 5.5:** la base de dades de codi obert més utilitzada, ens permetrà crear una base de dades SQL de manera senzilla on emmagatzemar les dades més estàtiques de la nostra aplicació.
- **PHP 5.3:** llenguatge de programació interpretat especialment dissenyat per a l'entorn web. Proporciona un conjunt de funcions que ens permetran accedir ràpidament a la base de dades MySQL.
- **Shell script:** llenguatge de programació interpretat propi dels sistemes Unix, utilitzat normalment per a tasques administratives relacionades amb el sistema.
- **Biblioteca Simple HTML DOM 1.5:** biblioteca en PHP que facilita el *web scraping* permetent recórrer el DOM de les diferents pàgines.
- **HTML:** llenguatge de marcatge en què estan escrites la majoria de pàgines web.

A l'apèndix 1 trobareu els detalls de com realitzar la instal·lació del sistema operatiu del servidor i tots els paquets de programari necessaris.

### 2.2.2. Aplicació Android

L'aplicació Android serà la que proporcionarà la interfície d'usuari i obtindrà les dades del servidor de dades.

En tractar-se d'una aplicació Android, el llenguatge de programació serà Java, llevat de la interfície que s'escriu en XML. Farem ús de diverses biblioteques que ens facilitaran algunes tasques.

- **Llenguatge de programació Java:** el llenguatge de programació Java, l'estàndard en Android, ens facilitarà la codificació si ja coneixem aquest llenguatge. És un llenguatge que s'executa en una màquina virtual, com mostrarem en següents apartats.
- **XML:** llenguatge de marcatge que permet definir estructures de dades. En el nostre cas l'utilitzarem per definir la interfície i les característiques de la nostra aplicació, fent ús de les etiquetes pròpies d'Android.
- **Biblioteca ActionBarSherlock 4.3:** facilita la utilització de la barra d'acció en dispositius Android anteriors a la versió 3.0.
- **Biblioteca HoloEverywhere 1.5+:** permet utilitzar els temes “Holo” (estàndard d'Android) en dispositius anteriors a la versió 4.0 d'Android, fent que totes les versions tinguin la mateixa interfície.
- **Biblioteca de suport per a versions anteriors d'Android:** permet utilitzar tecnologies de noves versions d'Android en versions anteriors del sistema operatiu.
- **Biblioteca Google Play Services:** permet utilitzar els mapes de Google Maps.
- **Altres biblioteques:** utilitzarem també les biblioteques ACRA (enviament d'informes d'errors), Google Analytics (estadístiques anònimes) i AdMob (monetització mitjançant publicitat).

### 2.3. La plataforma mòbil escollida: Android

Hem escollit la plataforma Android ja que és la plataforma líder pel que fa al mercat de dispositius mòbils. A més, tenint experiència amb Java i l'entorn de desenvolupament Eclipse, ens serà senzill adaptar-nos-hi.

#### 2.3.1. Funcionament del sistema operatiu

Android és un sistema operatiu de codi obert basat en un nucli Linux. Les aplicacions d'usuari s'executen dins d'una màquina virtual com en Java, però en aquest cas es tracta d'una implementació diferent realitzada per Android: la màquina virtual Dalvik. Això implica que no és compatible amb el “bytecode” de Java, tot i que sí que és possible compilar la majoria de biblioteques Java sota Android.

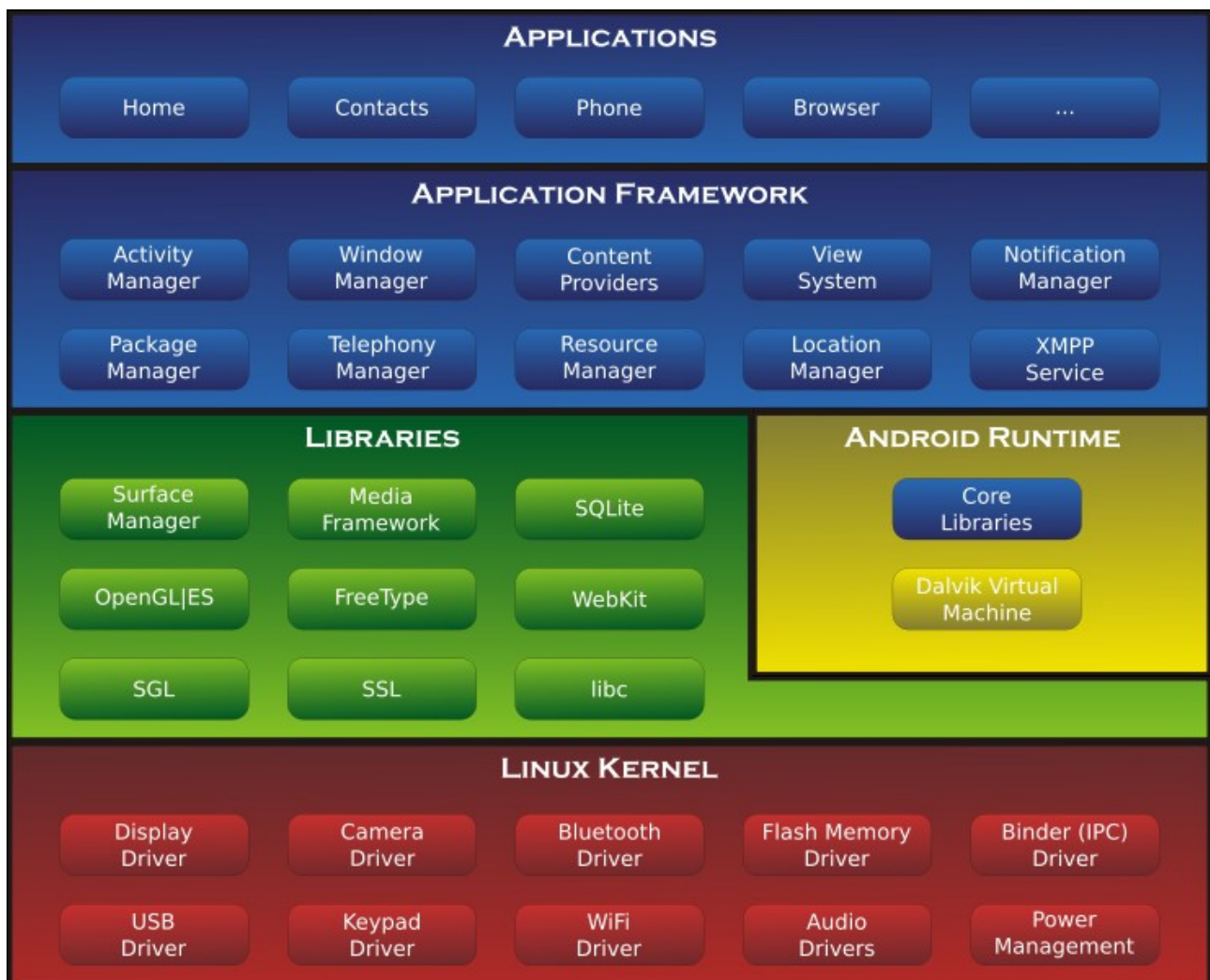
Les aplicacions s'executen cadascuna en un procés independent i amb la seva pròpia màquina virtual Dalvik. La gestió de memòria queda determinada pel nucli Linux.[9]

Per norma general, les aplicacions quan passen a segon pla no es tanquen sinó que queden en un estat suspès, tancant-se només en cas de requerir més memòria per obrir noves aplicacions.

A nivell de programació, Android abstrau tots els aspectes de hardware de manera que mai sol ser necessari realitzar programació a baix nivell.

L'entorn d'execució (“*runtime*”) d'Android on s'executen les aplicacions el formen el conjunt de la màquina virtual Dalvik i les biblioteques bàsiques de Java.

A continuació presentem un esquema general de l'arquitectura d'Android[10][11]:



### 2.3.2. Publicació d'aplicacions i seguretat

El mètode de publicació d'aplicacions en l'ecosistema Android és mitjançant els mercats d'aplicacions, és a dir, aplicacions que permeten mostrar llistes i conjunts d'altres aplicacions perquè els usuaris escullin quines volen descarregar. El més conegut i instal·lat per defecte en pràcticament la totalitat de dispositius Android és el Play Store de Google, que és el que utilitzarem en aquest projecte. No obstant, és possible publicar l'aplicació en diversos mercats alhora per obtenir més visibilitat.

Per tal de publicar una aplicació al Play Store, caldrà primer obtenir un compte de desenvolupador. Per fer-ho, és necessari únicament un compte gratuït de Google, acceptar l'acord de distribució i pagar una quota inicial per registrar-se (actualment, 25 dòlars)[12].

A continuació només cal carregar el fitxer d'aplicació (fitxer APK) i omplir el formulari amb la informació de l'aplicació (títol, descripció, captures de pantalla, icona, etc.), i en un temps

aproximat d'unes 3 hores ja estarà disponible al Play Store perquè es pugui descarregar des de gairebé tot el món.

D'altra banda, el Play Store i Android en general fan ús d'un sistema de seguretat per a les aplicacions basats en permisos concrets. D'aquesta manera, l'usuari ha d'acceptar expressament la llista de permisos sol·licitats per una aplicació per poder instal·lar-la. Això evita que els usuaris instal·lin aplicacions sospitoses. Per exemple, és sospitós que una aplicació que mostra la informació meteorològica demani el permís “afegir, modificar i eliminar adreces d'interès del navegador”.

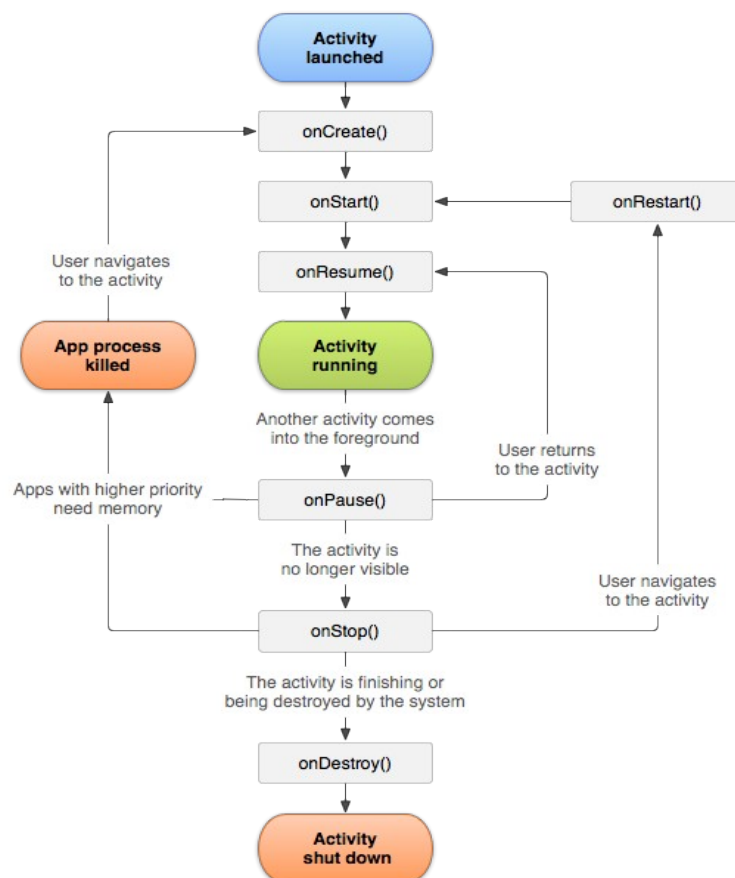
No obstant, aquest sistema també pot comportar problemes fent que els usuaris no instal·lin l'aplicació si no comprenen exactament per a què serveix un permís. Per exemple, per definir un to de trucada per a un contacte específic, cal el permís “llegir i escriure dades de contactes”, la qual cosa pot fer pensar a l'usuari que l'aplicació extraurà totes les dades dels seus contactes i les enviarà al desenvolupador de l'aplicació.

El conjunt de permisos necessaris cal definir-lo en el manifest de l'aplicació (un fitxer XML amb un format concret). En cas de no sol·licitar un permís i fer ús de la funcionalitat que requereix el permís, el sistema força el tancament de l'aplicació llançant un error.

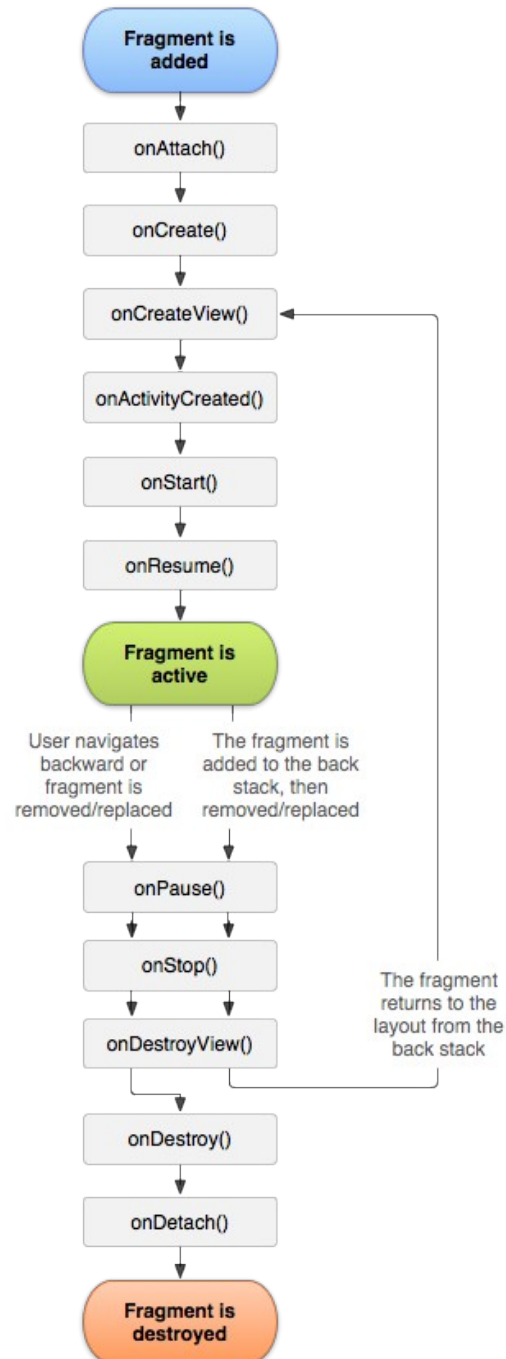
### 2.3.3. Components bàsics d'una aplicació

Cal destacar diversos components bàsics dintre d'una aplicació, que són els següents:

- **Activitat (“Activity”)**: equival a una pantalla de l'aplicació, i és l'encarregada de mostrar la interfície. Ve regida pel cicle de vida de les activitats, que és el següent[13]:



- **Fragment:** concepte introduït amb la versió 3.0 d'Android i que facilita la reutilització de parts de la interfície per modelar pantalles diferents en mòbils i tauletes. Té un funcionament similar al de les activitats, però es regeix pel seu propi cicle de vida independent, que és aquest[14]:



- **Manifest:** es tracta d'un fitxer XML (**AndroidManifest.xml**) que defineix les dades bàsiques de l'aplicació: paquet Java de l'aplicació, versió, nom, permisos necessaris, components de l'aplicació, versions d'Android suportades, etc.
- **Intent:** Crida del sistema que permet enllaçar les dades de diferents aplicacions. És el

component essencial que permet que les aplicacions intercanviïn dades entre elles, comparteixin elements, etc.

- **Tasca asíncrona (“AsyncTask”)**: Classe proporcionada per Android que facilita la creació de tasques asíncrones, executant-ne el codi principal en un fil independent, i retornant al fil principal quan és necessari. La seva estructura habitual és la següent:

```
class LaMevaAsyncTask extends AsyncTask<Params, Progress, Result> {
    protected Result doInBackground(Params... params) {
        //aquest mètode s'executa en un fil a part.
        //aquí fem tota la feina en segon pla.
        //si volem publicar un resultat temporal
        //(per exemple, per mostrar una barra de progrés),
        //cridarem publishProgress(progress);
        return result;
    }
    protected void onProgressUpdate(Progress... progress) {
        //aquest mètode s'executa al fil de la IU
        //actualitzar la IU amb un resultat temporal
    }
    protected void onPostExecute(Result result) {
        //aquest mètode s'executa al fil de la IU
        //actualitzar la IU amb el resultat final
    }
}
```

Els paràmetres **Params**, **Progress** i **Result** són tipus genèrics, i els podem canviar per les classes que creguem apropiades (per exemple, **Void**, **Integer** o **BusStation**).

Hi ha altres components bàsics (serveis, proveïdors de contingut, etc.) però no creiem necessari detallar-los perquè no els utilitzarem en aquest projecte.

### 2.3.4. Entorn de desenvolupament

L'entorn de desenvolupament estàndard per a Android és Eclipse amb l'extensió ADT (Android Development Tools). En el nostre cas, utilitzarem Eclipse 4.2.2 i ADT 22.

L'extensió ADT proporciona les eines necessàries per descarregar els diferents SDK d'Android i proporciona una interfície per configurar els emuladors de les diferents versions del sistema de manera senzilla.

Un altre component d'aquesta extensió permet editar la interfície de manera visual, generant automàticament l'XML necessari, tot i que sempre cal fer-hi retocs manualment per acabar-lo d'ajustar.

Finalment, aquestes eines ens permetran signar l'aplicació amb la nostra clau de desenvolupador i exportar-la en format APK per poder publicar-la al Play Store.

A l'apèndix 2 trobareu els detalls necessaris per realitzar la instal·lació de l'entorn de

desenvolupament i configurar-hi el codi de l'aplicació.

### **2.3.5. Fragmentació**

La fragmentació és un terme sovint utilitzat en Android per referir-se a la dificultat de programació d'aplicacions per a Android, donat que hi ha molts dispositius diferents. Per exemple, hem de tenir en compte en dissenyar les pantalles que es poden mostrar en pantalles molt petites o molt grans, o amb diferents densitats. Però el principal problema és la fragmentació creada per les diferents versions d'Android. Mentre que l'última versió d'Android és la 4.2, la majoria d'usuaris encara utilitzen la versió 2.3, ja sigui perquè el fabricant del dispositiu no l'ha actualitzat a versions posteriors o perquè el telèfon no té prou potència per executar noves versions d'Android.

En aquest projecte fem ús de diverses biblioteques per facilitar la consistència en totes les versions d'Android, encara que sigui antigues: la biblioteca de compatibilitat amb versions anteriors d'Android, ActionBarSherlock i HoloEverywhere. D'aquesta manera, suportem Android des de la versió 2.2 (el 98,4% del mercat[15]) fins a l'última versió i en tots els dispositius l'aplicació es veu amb el mateix estil.

### 3. Disseny de l'aplicació

A continuació detallarem el disseny tant a nivell tecnològic com a nivell de funcionalitat i d'interfície.

#### 3.1. Funcionalitat i interfície d'usuari

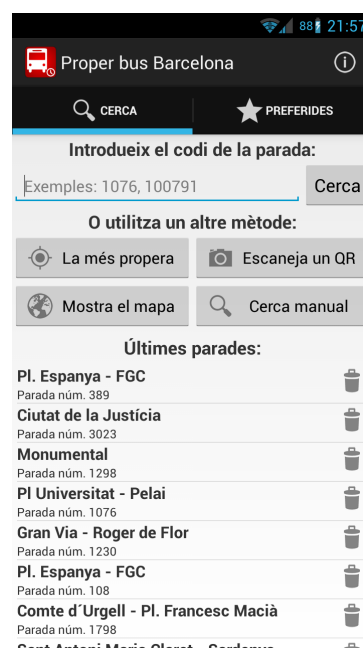
La funcionalitat més bàsica i ràpida cal que sigui la consulta per codi de parada. Per tant, hem decidit emprar una estructura d'aplicació clàssica d'Android, amb dues pestanyes: cerca i preferides. Es pot canviar de pestanya lliscant amb el dit o prement el títol de la pestanya.

##### 3.1.1. Pantalla de cerca

A la pestanya de cerca, s'hi mostrarà un camp per introduir-hi el codi de la parada, limitat a només xifres, juntament amb un botó “Cerca” que realitzarà la consulta.

A part d'aquesta opció, s'ofereixen diferents opcions útils per si no es coneix el codi de parada: “la més propera” (geolocalitza per GPS o xarxes mòbils i mostra directament la parada més propera), “escaneja un QR” (obre la càmera per escanejar el codi QR de les parades), “mostra el mapa” (mostra un mapa geolocalitzat amb totes les parades) i “cerca manual” (obre un formulari de cerca de parades amb diferents criteris).

Per últim, a la part inferior s'hi mostren les últimes parades consultades, amb un botó d'eliminar per a cadascuna, per si es volen esborrar de la llista. Si la llista és buida (per exemple, quan s'executa per primera vegada l'aplicació), s'hi mostra informació sobre on es pot trobar el codi de parada.



##### 3.1.2. Pantalla de preferides

A la pestanya de parades preferides, s'hi mostren les parades de bus que l'usuari ha marcat com a preferides (es fa des de la pantalla d'una parada).

Es permet reanomenar les parades, per tal de donar-los noms que permetin identificar-les més fàcilment (per exemple: “Feina” o “Casa”), així com esborrar-les dels preferits, mitjançant els dos botons de què disposa cada element.

Les parades es consulten simplement prement-les.

##### 3.1.3. Pantalla d'una parada

A la pantalla de consulta d'una parada, s'hi mostra principalment la informació de les properes línies en arribar, ordenades per temps restant.

Cada línia de la llista mostra: identificador de la línia (amb el color





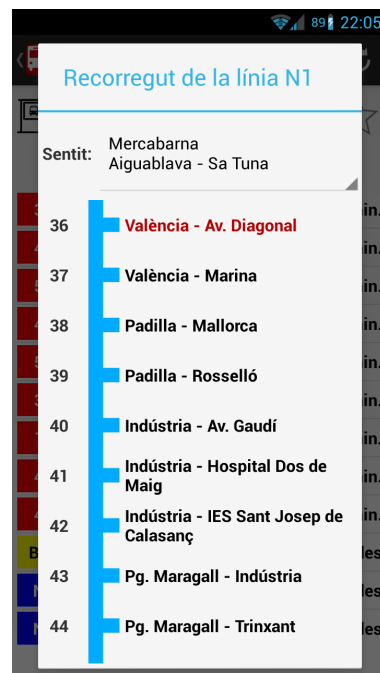
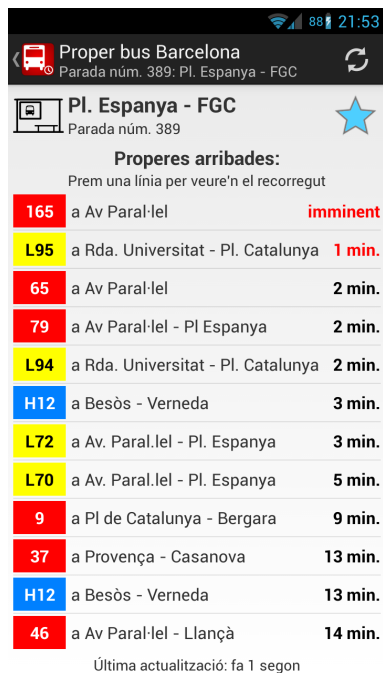
de l'operadora), destinació, i temps restant. Si es fa clic sobre la línia, se'n mostra el recorregut.

A la part superior de la pantalla s'hi indica la parada escollida, incloent-ne el nom, codi i tipus de parada (amb una icona). També hi ha present el botó per afegir-la/esborrar-la dels preferits.

Les dades es poden refrescar prement el botó de refresc de la barra d'accions, i l'última actualització es mostra a la part inferior de la pantalla.

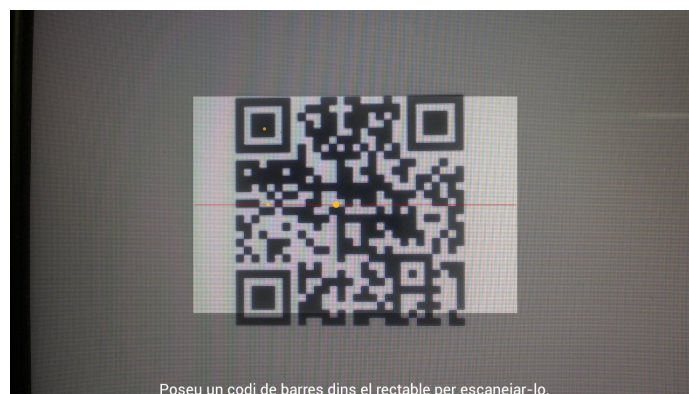
La pantalla que permet veure el recorregut del bus mostra una llista amb totes les parades, on l'actual apareix pintada de color vermell.

Si la línia té diversos sentits, es pot escollir el sentit amb l'*spinner* de la part superior de la pantalla.



### 3.1.4. Pantalla d'escaneig per QR

Aquesta pantalla mostra el lector de QR del dispositiu per tal d'escanejar el codi QR de la parada. L'aplicació reconeix els codis QR de les parades de TMB, de l'AMB i els codis de barres interns de les marquesines de TMB. Una vegada escanejat el codi QR, s'obre directament la pantalla amb la informació de la parada.

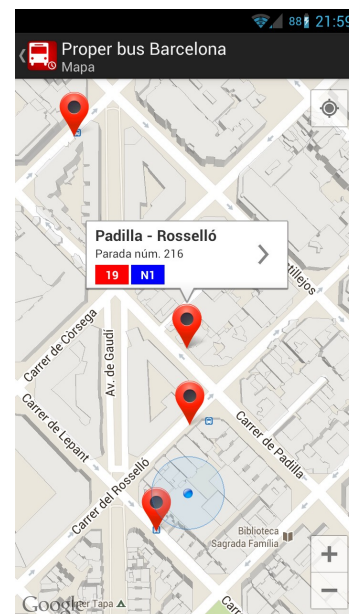


### 3.1.5. Pantalla del mapa

La pantalla del mapa utilitza un mapa de Google Maps i geolocalitza l'usuari al punt on estigui ubicat actualment. Si no és possible geolocalitzar-lo, es mostra centrat a la Plaça de Catalunya de Barcelona, però amb un nivell d'ampliació reduït perquè l'usuari es pugui moure fins on és.

Quan el nivell d'ampliació és prou gran, es mostren les icones que indiquen on hi ha cada parada de bus. Si el nivell d'ampliació no és prou gran, es mostra el missatge: "Amplia més per veure les parades". El motiu d'això és que en cas de mostrar tots els punts de les parades disminueix el rendiment del dispositiu i, a més, es fa difícil prémer sobre un element sense equivocar-se.

En prémer sobre una de les parades, es mostra un globus que indica el nom de la parada, el seu codi i les línies que hi paren, distingides amb el color de l'operadora. Si es prem al globus, s'accedeix a la pantalla de consulta d'aquella parada, igual que si se n'hagués introduït el codi a mà.



### 3.1.6. Pantalla de cerca manual

Aquesta pantalla permet realitzar una cerca de parades de bus a partir de diversos paràmetres, com ara: nom de la parada, població, línia que hi passa i empresa operadora de la línia.

El nom de la parada és un camp de text, mentre que la resta són *spinners*.

Una vegada es prem el botó per fer la cerca, s'obre una nova pantalla amb els resultats. Si es fa clic en algun d'ells, s'obre la pantalla de la parada seleccionada.

Es mostren com a màxim 100 resultats, si se sobrepassa el màxim, es mostra un missatge indicant que cal restringir més la cerca.

Resultats de la cerca:	
Hi ha massa resultats, se'n mostren només 100. Intenta reduir la cerca.	
Àngel Guimerà - Josep Argemí	Parada núm. 100121
Av Tibidabo - Josep Maria Florensa	Parada núm. 1358
Av. Alfons XIII - Josep M <sup>a</sup> de Sagarra	Parada núm. 2995
Av. de Sarrià - Josep Tarradellas	Parada núm. 630
Av. Diagonal - Josep Pla	Parada núm. 1890
Av. Fabregada - Josep Pla	Parada núm. 107100
Av. Generalitat - Av. Josep Tarradellas	Parada núm. 100556
Av. Isidre Martí - Josep Rodoreda	Parada núm. 100688
Av. Josep Tarradellas - "ATRIUM"	Parada núm. 108830
Av. Josep Tarradellas - "Biblioteca"	Parada núm. 107315
Av. Josep Tarradellas - "CAP Maria Bernades"	Parada núm. 106633
Av. Josep Tarradellas - Av. Isabel la Catòlica	Parada núm. 109224
Av. Josep Tarradellas - Comte Borrell	

### 3.1.7. Pantalla “Quant a”

La pantalla “Quant a” mostra simplement la versió de l'aplicació, així com dades del desenvolupador i informació legal.

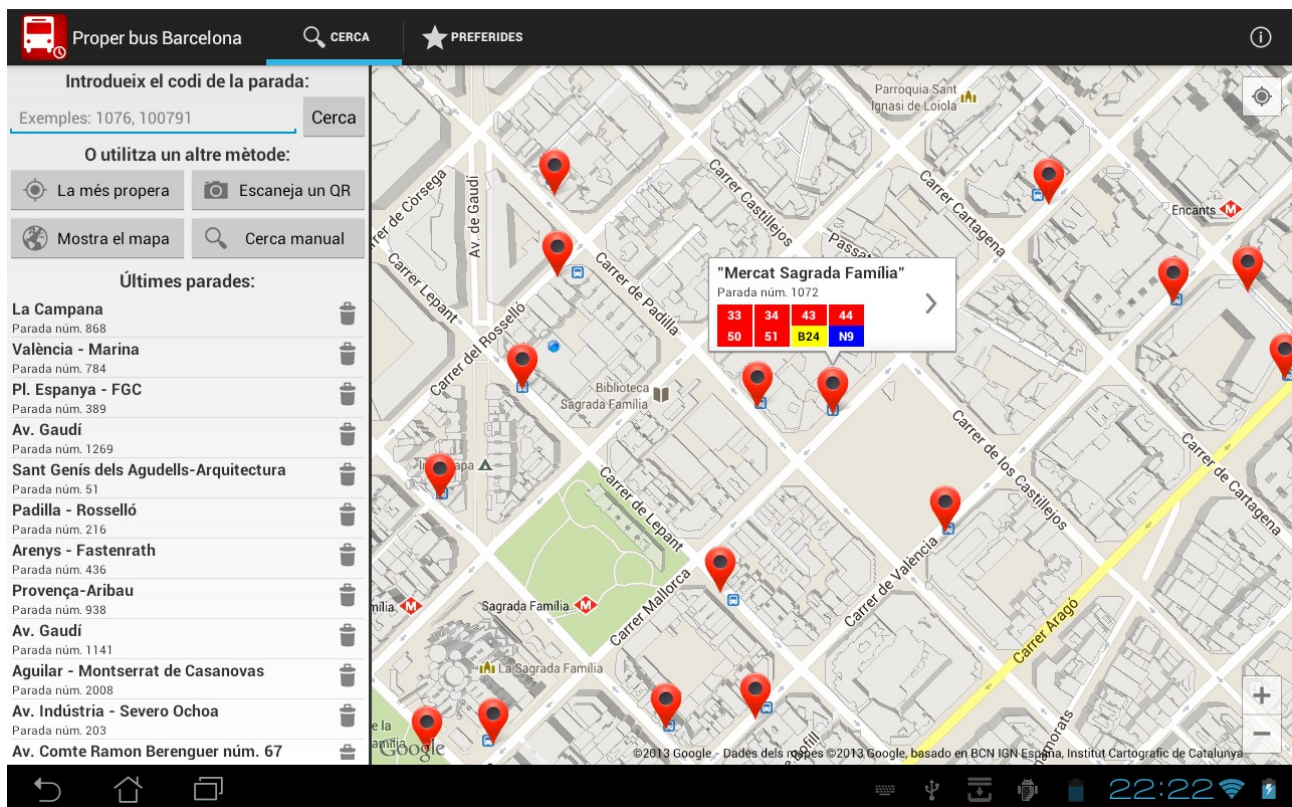
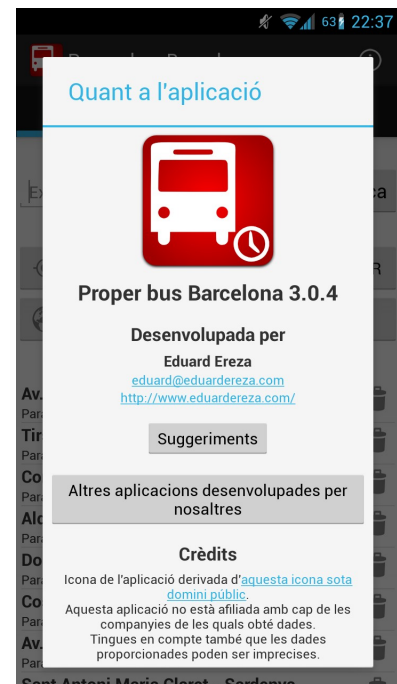
En aquesta pantalla es mostra el botó de “Suggeriments”, que permet enviar suggeriments de millora de l'aplicació o bé informar d'errors. En fer-hi clic, s'obre l'aplicació de correu electrònic de l'usuari amb un tema i destinatari predefinit.

### 3.1.8. Disseny per a tauletes

L'aplicació està dissenyada de manera que la interfície és diferent depenent de la mida de la pantalla del dispositiu.

Així, si el dispositiu és un telèfon mòbil, es mostren les pantalles que hem vist en aquest capítol. En canvi, si es tracta d'una tauleta de 7” o més, es manté la pantalla inicial amb dues pestanyes durant tota l'aplicació a la part esquerra de la pantalla, i les diferents pantalles es mostren a la part dreta de la pantalla.

A continuació se'n pot veure un exemple:



## 3.2. Estructura de l'aplicació

A continuació detallarem l'estructura de l'aplicació, tant a nivell de components lògics com a nivell de fitxers i codi.

### 3.2.1. Esquema general

L'aplicació es divideix en dos grans blocs: l'aplicació Android i els processos d'obtenció de dades, instal·lats en un servidor web.

L'aplicació Android mostra la interfície d'usuari i realitza les consultes contra el servidor de dades mitjançant una sèrie de peticions HTTP.

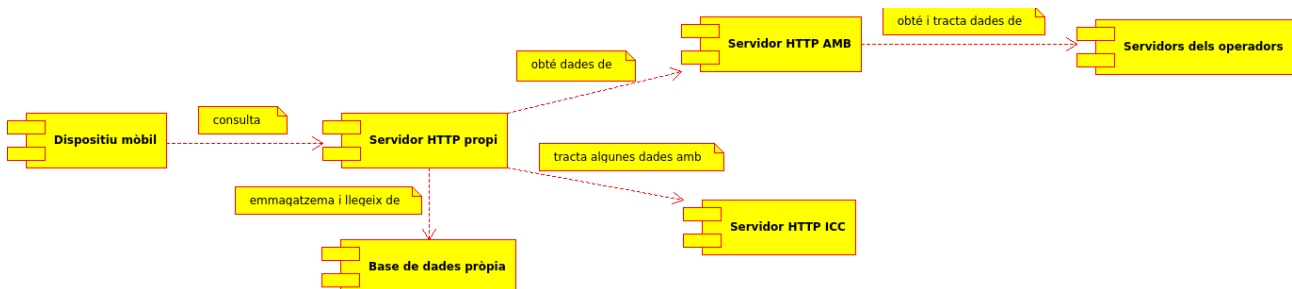
Els processos d'obtenció de dades obtenen dos tipus de dades:

- Dades en temps real (temps d'arribada dels autobusos a les parades)
- Dades estàtiques (línies de bus, parades, itineraris, mobiliari de les parades, etc.)

Pel que fa a les dades en temps real, s'obtenen mitjançant *web scraping*, fent una petició a la pàgina web de l'AMBTempsBus en el moment de la consulta (la qual internament consulta els sistemes dels diferents operadors), i creua les dades obtingudes amb altres dades de la base de dades.

Pel que fa a les dades estàtiques, s'obtenen cada nit de les diferents pàgines de l'AMB mitjançant *web scraping*, i s'emmagatzemen amb una estructura concreta dins una base de dades MySQL. Dins d'aquest procés, es converteixen les coordenades proporcionades per l'AMB a coordenades latitud/longitud, mitjançant peticions web a un convertidor de coordenades de l'Institut Cartogràfic de Catalunya.

A continuació mostrem un diagrama dels diferents components i quina relació tenen:



### 3.2.2. Aplicació Android

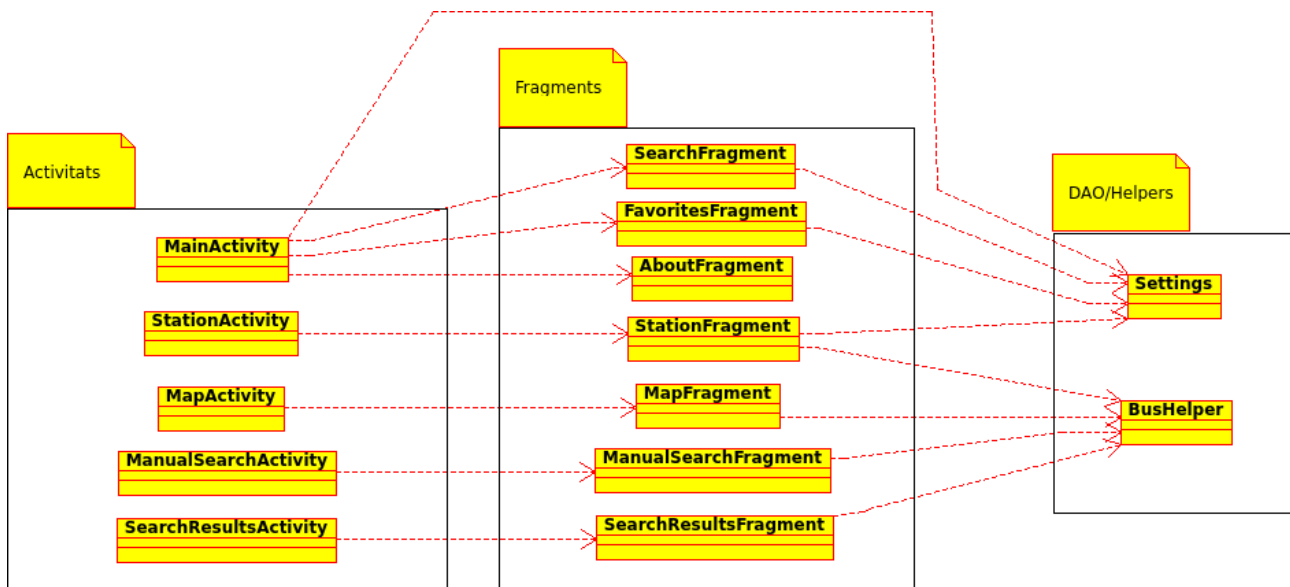
A continuació detallem els diferents elements que componen l'aplicació Android i com estaran estructurats.

#### 3.2.2.1. Classes de l'aplicació i interacció

L'esquema general de les classes principals i les diferents interaccions queden expressades en el següent diagrama. Hi podem distingir tres capes principals:

- La capa d'activitats correspon a les activitats d'Android, i és qui s'encarrega de gestionar els fragments i com es mostren (per exemple, en el cas de les tauletes).
- La capa de fragments és l'encarregada de la major part de la visualització de les pantalles, així com de la lògica associada a la seva funcionalitat. Els fragments sovint contenen classes internes (*adapters*, tasques asíncrones, etc.) que els ajuden a fer les seves tasques de negoci.
- La capa DAO i *helpers* consta de dues classes: la classe **Settings** és l'encarregada d'emmagatzemar la configuració de l'aplicació (parades preferides, últimes consultades,

etc.). La classe **BusHelper**, d'altra banda, és l'encarregada d'obtenir les dades via HTTP des del servidor. Aquesta classe inclou també altres mètodes útils en punts concrets de l'aplicació.



Hi ha moltes classes que no queden reflectides en aquest diagrama (*beans*, utilitats concretes, etc.), per tant, en farem un breu resum a continuació:

`cat.ereza.properbusbcn`

- **beans**
  - **BusLineAndTime**: *Bean* que conté les dades de línia i temps (per a la llista de línies)
  - **BusStation**: *Bean* que conté el nom, id, i línies d'una parada de bus
  - **RouteElement**: *Bean* que conté l'element d'una ruta (identificador i nom de la parada)
  - **Route**: *Bean* que conté una ruta (conjunt d'elements, origen i destinació)
- **fragments**
  - **AboutFragment**: Fragment de la pantalla “Quant a”
  - **DeleteFavoriteStationDialogFragment**: Fragment del diàleg “Vols eliminar aquest preferit?”
  - **DeleteLatestStationDialogFragment**: Fragment del diàleg “Vols eliminar aquesta última parada?”
  - **FavoritesFragment**: Fragment dels preferits
  - **ManualSearchFragment**: Fragment de la cerca manual
  - **MapFragment**: Fragment del mapa
  - **RenameFavoriteStationDialogFragment**: Fragment que permet reanomenar una parada preferida
  - **RouteFragment**: Fragment que mostra la ruta d'una línia de bus
  - **SearchFragment**: Fragment que permet realitzar la consulta d'una parada de diferents



maneres (pantalla inicial)

- | — **SearchResultsFragment**: Fragment que mostra els resultats de la cerca manual
- | — **ShowLoadErrorOnMapDialogFragment**: Fragment del diàleg “S'ha produït un error en carregar el mapa”.
- | — **StationFragment**: Fragment de la parada de bus
- | — **globals**
  - | — **Globals**: Conté les constants globals de l'aplicació
- | — **MainActivity**: Activitat principal, inici de l'aplicació
- | — **ManualSearchActivity**: Activitat que conté el fragment de cerca manual
- | — **MapActivity**: Activitat que conté el fragment del mapa
- | — **MyFragmentActivity**: Activitat de la qual hereten totes les activitats de l'aplicació, per si fos necessari utilitzar algun mètode comú en totes elles. Actualment no s'utilitza.
- | — **ProperBusApplication**: Classe d'aplicació, necessària per utilitzar ACRA.
- | — **SearchResultsActivity**: Activitat que conté el fragment dels resultats de la cerca manual
- | — **StationActivity**: Activitat que conté el fragment de la parada de bus
- | — **util**
  - | — **BusHelper**: Classe que obté les dades del servidor via HTTP i que conté altres funcions útils en algun punt de l'aplicació
  - | — **NaturalOrderComparator**: Classe que permet fer una ordenació natural de les línies de bus a la pantalla de cerca manual.
  - | — **Settings**: Classe que llegeix i desa la configuració de l'aplicació (parades preferides, etc.)
  - | — **Timer**: Classe útil per comprovar els temps d'execució. Només útil en fase de desenvolupament.
  - | — **Tracker**: Classe que envia estadístiques anònimes a Google Analytics. Es crida des de diversos punts de l'aplicació
  - | — **ViewHolder**: Classe que s'utilitza per reciclar les vistes en les diferents llistes.

**com.google.zxing.integration.android**

- | — **IntentIntegrator**: Classe que permet integrar el lector de QR “Barcode Scanner”
- | — **IntentResult**: Resultat d'un escaneig amb “Barcode Scanner”

### 3.2.2.2. Estructura dels “layouts”

Els *layouts* o dissenys de pantalles són fitxers XML que defineixen els diferents components de cada pantalla. En casos concrets, cal definir-ne de diferents segons la resolució o densitat de pantalla, per exemple, per mostrar un disseny diferent si l'aplicació s'executa en una tauleta.

Aquesta és l'estructura de *layouts* que seguirem en el projecte, i la descripció de cadascun d'ells:

**res**

- └─ [...]
- └─ **layout**: Conté els dissenys per defecte
  - └─ **about\_fragment.xml**: Pantalla “Quant a”
  - └─ **activity\_main.xml**: Pantalla principal
  - └─ **activity\_manual\_search.xml**: Activitat de cerca manual (conté el fragment)
  - └─ **activity\_map.xml**: Activitat del mapa (conté el fragment)
  - └─ **activity\_search\_results.xml**: Activitat de resultats de cerca (conté el fragment)
  - └─ **activity\_station.xml**: Activitat de la parada de bus (conté el fragment)
  - └─ **favorites\_fragment.xml**: Pantalla de preferides
  - └─ **favorite\_stations\_list.xml**: Element de la llista de preferides
  - └─ **latest\_stations\_list.xml**: Element de la llista d'últimes parades
  - └─ **line\_item.xml**: Text de l'identificador de línia de bus
  - └─ **lines\_list.xml**: Element de la llista de línies
  - └─ **manual\_search\_fragment.xml**: Pantalla de cerca manual
  - └─ **map\_info\_window.xml**: Globus d'informació del mapa
  - └─ **multiline\_spinner\_dropdown\_item.xml**: Estil per a l'*spinner* de HoloEverywhere
  - └─ **multiline\_spinner\_item.xml**: Estil per a l'element de l'*spinner* de HoloEverywhere
  - └─ **progress\_wheel.xml**: Barra de progrés indeterminat
  - └─ **route\_fragment.xml**: Pantalla de la ruta
  - └─ **routes\_list.xml**: Element de la llista de la ruta
  - └─ **search\_fragment.xml**: Pantalla de cerca (inicial)
  - └─ **search\_results\_fragment.xml**: Pantalla de resultats de la cerca manual
  - └─ **station\_fragment.xml**: Pantalla de la parada de bus
  - └─ **stations\_results\_list.xml**: Element de la llista de resultats de la cerca manual
- └─ **layout-land**: Dissenys per a dispositius en horitzontal
  - └─ **search\_fragment.xml**: Posa els quatre botons principals en una sola línia
- └─ **layout-sw600dp**: Dissenys per a dispositius amb 600dp o més (tauletes de 7")
  - └─ **activity\_main.xml**: Pantalla dividida entre cerca/preferits i resta (mode tauleta)
- └─ **layout-sw600dp-land**: Dissenys per a dispositius amb 600dp o més en horitzontal
  - └─ **search\_fragment.xml**: Disseny per defecte, per reemplaçar l'específic “layout-land”
- └─ **layout-sw720dp**: Dissenys per a dispositius amb 720dp o més (tauletes de >7")
  - └─ **activity\_main.xml**: Igual que layout-sw600dp però amb 350dp a la part esquerra
- └─ **layout-sw720dp-land**: Dissenys per a dispositius amb 720dp o més en horitzontal

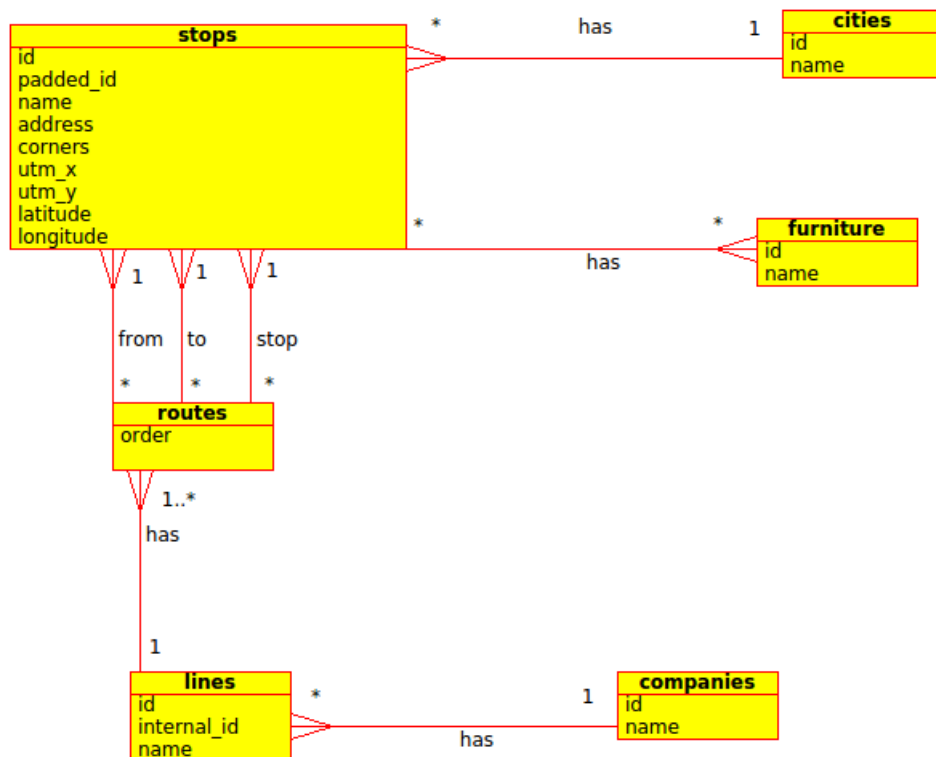
- |   └─ `search_fragment.xml`: Disseny per defecte, per reemplaçar l'específic “layout-land”
- |─ `layout-xlarge`: Dissenys per a dispositius amb pantalla extragran (Android <3.0)
- |   └─ `activity_main.xml`: Igual que layout-sw720dp
- └─ `layout-xlarge-land`: Dissenys per a dispositius amb pantalla extragran en horitzontal
  - └─ `search_fragment.xml`: Igual que layout-sw720dp-land

### 3.2.3. Processos d'obtenció de dades

Els processos d'obtenció de dades són els encarregats de mantenir actualitzada la base de dades, i de respondre a les peticions HTTP de l'aplicació Android.

#### 3.2.3.1. Estructura de base de dades

L'estructura de la base de dades és la següent:



#### Taula “cities”:

Conté les ciutats a les quals corresponen les parades de bus.

<u>Camp</u>	<u>Tipus</u>	<u>Nul?</u>	<u>Descripció</u>
<u>id</u>	VARCHAR(255)	No	Identificador de la ciutat
name	VARCHAR(255)	Sí	Nom de la ciutat



**Taula “companies”:**

Conté les empreses a les quals corresponen les línies de bus.

<b><u>Camp</u></b>	<b><u>Tipus</u></b>	<b><u>Nul?</u></b>	<b><u>Descripció</u></b>
<u>id</u>	VARCHAR(255)	No	Identificador de l'empresa
name	VARCHAR(255)	Sí	Nom de l'empresa

**Taula “furniture”:**

Conté el mobiliari de cada parada.

<b><u>Camp</u></b>	<b><u>Tipus</u></b>	<b><u>Nul?</u></b>	<b><u>Descripció</u></b>
<u>id</u>	VARCHAR(255)	No	Identificador de l'element de mobiliari
name	VARCHAR(255)	Sí	Nom de l'element de mobiliari

**Taula “lines”:**

Conté les línies de bus.

<b><u>Camp</u></b>	<b><u>Tipus</u></b>	<b><u>Nul?</u></b>	<b><u>Descripció</u></b>
<u>id</u>	VARCHAR(255)	No	Identificador de la línia
internal_id	INTEGER	Sí	Identificador intern de la línia a la web de l'AMB
name	VARCHAR(255)	Sí	Nom de la línia a la web de l'AMB
company_id	VARCHAR(255)	Sí	Identificador de l'empresa operadora

**Taula “routes”:**

Conté els itineraris de cada línia de bus per a cada itinerari (origen-final).

<b><u>Camp</u></b>	<b><u>Tipus</u></b>	<b><u>Nul?</u></b>	<b><u>Descripció</u></b>
<u>line_id</u>	VARCHAR(255)	No	Identificador de la línia
<u>origin_id</u>	INTEGER	No	Identificador de la parada origen del recorregut
<u>direction_id</u>	INTEGER	No	Identificador de la parada destinació del recorregut
<u>order</u>	INTEGER	No	Ordre dins de l'itinerari de la parada indicada
<u>stop_id</u>	INTEGER	Sí	Identificador d'una parada de l'itinerari

**Taula “stops”:**

Conté les parades de bus.

<u>Camp</u>	<u>Tipus</u>	<u>Nul?</u>	<u>Descripció</u>
<u>id</u>	INTEGER	No	Identificador de la parada
padded_id	CHAR(6)	Sí	Identificador de la parada, omplert amb zeros a l'esquerra si és necessari
name	VARCHAR(255)	Sí	Nom de la parada
address	VARCHAR(255)	Sí	Adreça de la parada
corners	VARCHAR(255)	Sí	Cantonades on se situa la parada
city_id	VARCHAR(255)	Sí	Ciutat on se situa la parada
utm_x	DECIMAL(11,4)	Sí	Coordenada UTM X de la parada
utm_y	DECIMAL(11,4)	Sí	Coordenada UTM Y de la parada
latitude	DECIMAL(12,9)	Sí	Latitud de la parada
longitude	DECIMAL(12,9)	Sí	Longitud de la parada

**Taula “rel\_stops\_furniture”:**

Conté la relació M:N entre les parades i el seu mobiliari.

<u>Camp</u>	<u>Tipus</u>	<u>Nul?</u>	<u>Descripció</u>
<u>stop_id</u>	INTEGER	No	Identificador de la parada
<u>furniture_id</u>	VARCHAR(255)	No	Identificador de l'element de mobiliari

**3.2.3.2. Estructura de directoris**

L'estructura de directoris i fitxers és la següent

```
proper_bus_bcn
├── checker
│   └── page_checker.sh
├── cities_list.php
├── companies_list.php
├── config.php
├── fetcher
│   ├── config.php
│   ├── current_db.sql
│   ├── current_log.txt
│   ├── fetch_all_data.php
│   ├── fetch_all_data.sh
│   ├── previous_db.sql
│   └── proper_bus_bcn_mysql.sql
└── libs
```

```
|   └─ simple_html_dom.php
|─ lines_list.php
|─ nearest_stop.php
|─ routes_list.php
|─ search_stop.php
|─ stop_info.php
└─ stops_list.php
```

Els fitxers `config.php` contenen les credencials de connexió a base de dades.

El fitxer `simple_html_dom.php` pertany a la biblioteca Simple HTML DOM.

El fitxer `proper_bus_bcn_mysql.sql` conté l'estructura de base de dades inicial del projecte.

Els fitxers `previous_db.sql` i `current_db.sql` són exportacions de base de dades anterior i posterior, respectivament, a l'execució del procés “`fetch_all_data`”, i `current_log.txt` conté el registre de l'última execució.

La resta de fitxers corresponen als processos i estan detallats en el següent punt.

### 3.2.3.3. Processos

A continuació exposem una breu descripció de cada procés del sistema. Trobareu el detall d'implementació de cada procés a l'apartat 4.

#### **Procés “`fetch_all_data`”:**

Aquest procés és l'encarregat d'obtenir cada nit les dades estàtiques de parades, línies, ciutats, empreses, rutes i mobiliari.

S'executa a través de `fetch_all_data.sh`, el qual crida `fetch_all_data.php` i comprova que s'hagi executat correctament. En cas que no hagi estat així, o si hi ha canvis en les dades obtingudes, envia un correu a l'administrador.

És llançat des d'una tasca programada amb cron i no és accessible via URL.

#### **Procés “`cities_list`”:**

Aquest procés retorna la llista de ciutats que hi ha a la base de dades. S'hi accedeix mitjançant una URL.

#### **Procés “`companies_list`”:**

Retorna la llista d'empreses que hi ha a la base de dades. S'hi accedeix mitjançant una URL.

#### **Procés “`lines_list`”:**

Retorna la llista de línies que hi ha a la base de dades. S'hi accedeix mitjançant una URL.

#### **Procés “`nearest_station`”:**

Donats un paràmetres de latitud i longitud, calcula i retorna l'identificador de la parada més propera.

S'hi accedeix mitjançant una URL.

### **Procés “routes\_list”:**

Donada una línia, retorna tots els itineraris de la línia. S'hi accedeix mitjançant una URL.

### **Procés “search\_stop”:**

Donats un nom, empresa, línia i ciutat (poden ser buits), realitza una cerca i retorna les primeres 100 parades coincidents. S'hi accedeix mitjançant una URL.

### **Procés “stop\_info”:**

Donat un codi de parada, retorna la llista de properes arribades de busos, juntament amb la seva direcció i temps, si són calculables. Si el temps no està disponible, retorna un temps negatiu.

S'hi accedeix mitjançant una URL.

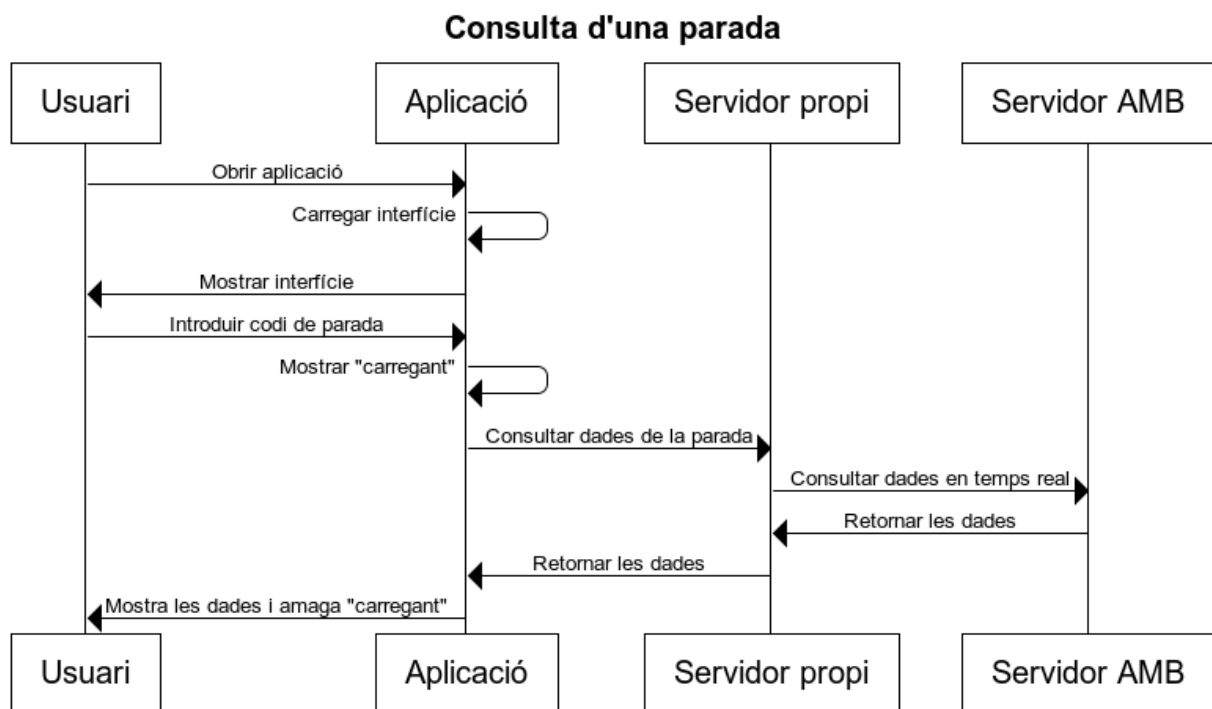
### **Procés “stops\_list”:**

Retorna la llista de totes les parades amb les seves coordenades. S'hi accedeix mitjançant una URL.

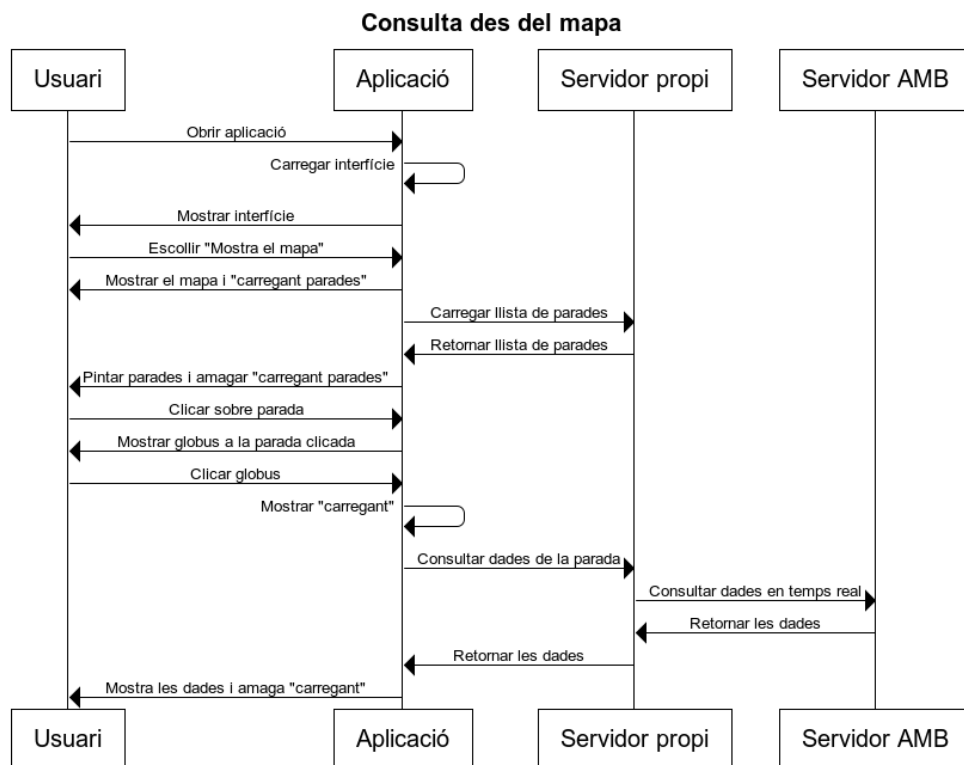
## **3.3. Casos d'ús i diagrames de seqüència**

A continuació mostren els diagrames de seqüència dels casos d'ús més complexos de l'aplicació, que corresponen als diferents botons de la pantalla inicial.

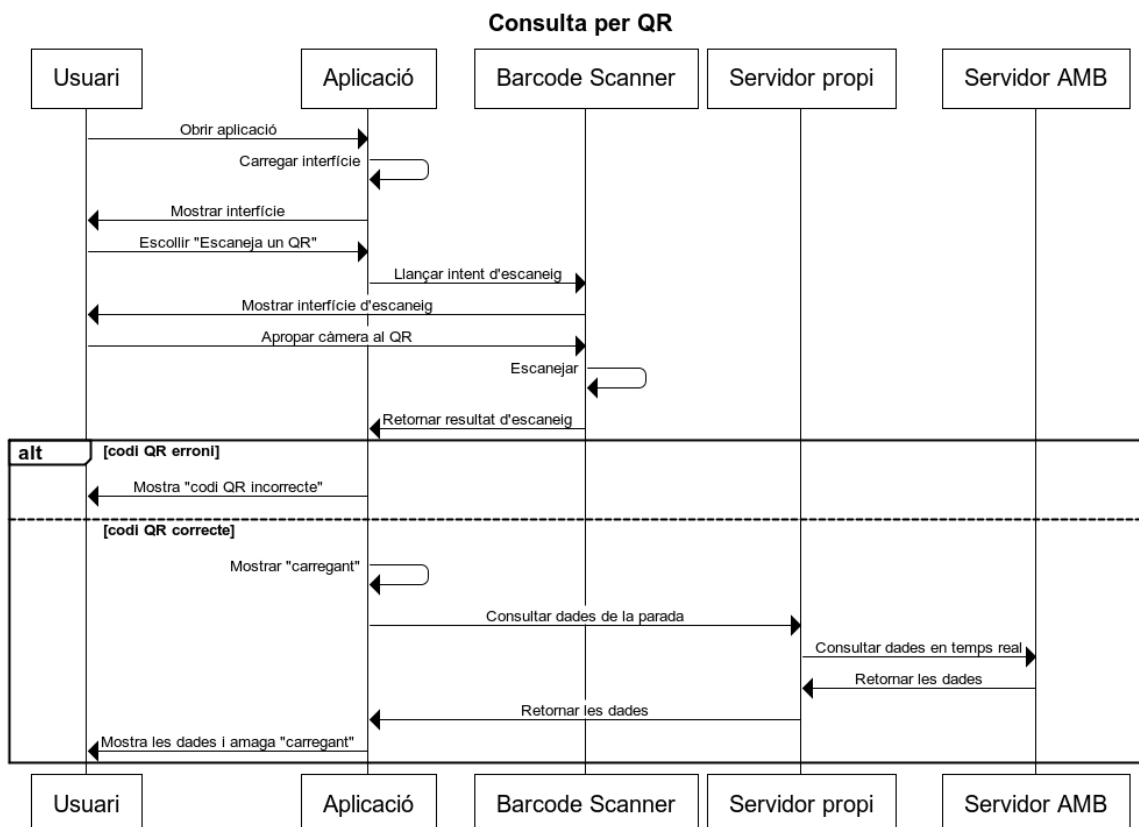
### **3.3.1. Cas d'ús “consulta d'una parada”**



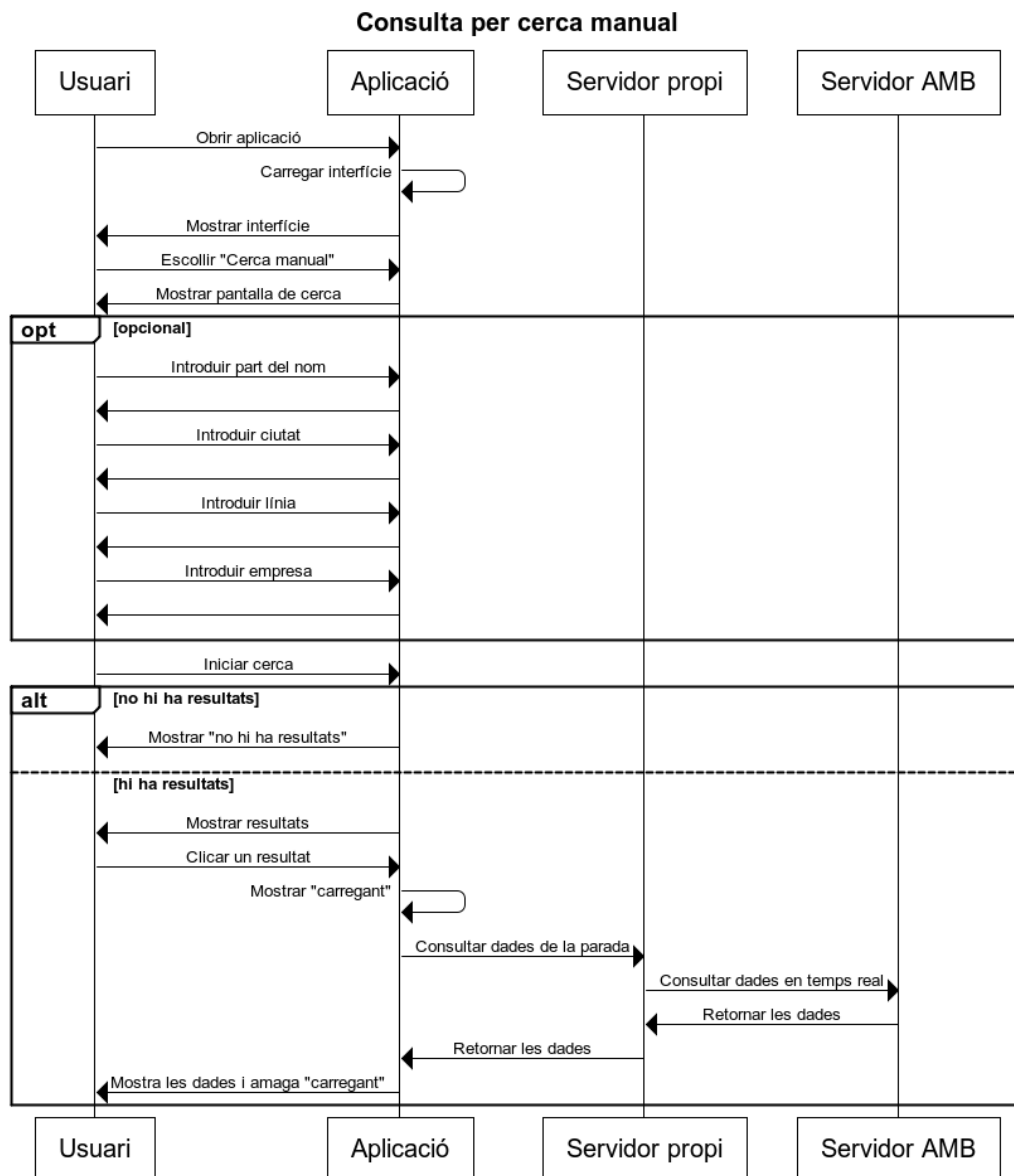
### 3.3.2. Cas d'ús “consulta des del mapa”



### 3.3.3. Cas d'ús “consulta per QR”



### 3.3.4. Cas d'ús “consulta per cerca manual”



### 3.3.5. Altres casos d'ús

Altres casos d'ús que no hem cregut necessari especificar ja que són més senzills serien:

- Consulta d'una parada preferida fent-hi clic
- Consulta d'una última parada consultada fent-hi clic
- Consulta de la pantalla “Quant a”
- Refresc de la parada de bus
- Consulta d'una ruta fent clic a la línia

## 4. Aspectes d'implementació

Destaquem a continuació aspectes que considerem importants pel que fa a la implementació de les aplicacions, així com les dificultats trobades en el projecte i els límits de la implementació.

### 4.1. Implementació dels processos

Hem considerat apropiat destacar a grans trets la implementació dels processos d'obtenció de dades, ja que són la part més bàsica del projecte.

#### 4.1.1. Procés “fetch\_all\_data”

Aquest procés consisteix en un *shell script* que comprova la seva correcta execució i el procés pròpiament dit, implementat en PHP.

Pel que fa al *shell script*, realitza el següent:

1. Exporta la base de dades anterior a l'execució del procés, i la desa en el fitxer `previous_db.sql`:  

```
mysqldump --skip-comments --skip-extended-insert -u[usuari]
-p[contrasenya] proper_bus_bcn > previous_db.sql
```
2. Executa el procés i desa el seu registre a `current_log.txt`.
3. Si el codi de retorn del procés no és 0 (indica algun error), envia un correu a l'administrador indicant que s'ha produït un error que cal revisar.
4. En cas contrari, exporta la base de dades posterior a l'execució del procés, i la desa al fitxer `current_db.sql`, de la mateixa manera que al punt 1.
5. Compara els fitxers i si són diferents, genera un “diff” i l'envia en un correu a l'administrador perquè pugui identificar què ha canviat.

Pel que fa al procés pròpiament dit, realitza el següent per tal d'obtenir totes les dades:

1. Connecta a la base de dades i defineix la codificació a UTF-8 i deshabilita l'autocommit.
2. Esborra totes les taules mitjançant **DELETE**. Com que l'autocommit està deshabilitat, l'aplicació seguirà funcionant amb les dades anteriors fins que fem commit.
3. Obté la llista de totes les línies de bus. Això ho fa accedint a la següent URL:

`http://www.ambmobilitat.cat/Principales/BusquedaLinea.aspx?cab=1`

Aquesta URL correspon a la pàgina “Línies, horaris i plànols” de l'AMB. El paràmetre `cab=1` s'utilitza en accedir-hi des de mòbils, i el passem perquè d'aquesta manera no es passa la capçalera de la pàgina, per tant, estalviem ample de banda.

Les dades s'extreuen del “combo” de línies de bus (amb id `ddlLineas`), de manera que utilitzarem aquest codi:

```
$data = $html->find('select#ddlLineas option');
```

Per a cada línia, la insereix a la base de dades i aprofita la iteració de les dades per desar l'identificador intern i el codi de cada línia en una matriu anomenada `lines_array`.

4. Recorre la matriu `lines_array` i per a cada línia consulta la seva ruta d'anada i tornada. La ruta d'anada l'obté consultant la següent URL:

```
http://www.ambmobilitat.cat/Principales/TiraRecorrido.aspx?
cab=1&linea=[id_intern_de_la_linia]
```

Cal destacar que des de la web de l'AMB, no és possible accedir al recorregut de les línies de TMB des d'un navegador (ens redirigeixen a la web de TMB), però sí que el podem obtenir cridant la mateixa URL que per a la resta de línies amb l'identificador intern de la línia.

En la consulta de la ruta d'anada també desa a la base de dades l'empresa que opera la línia (s'obté del camp de text amb identificador `lblNombreEmpresaIda`) i la insereix a base de dades (amb un `REPLACE INTO`).

La ruta en si s'obté de la taula mostrada a la pàgina, de la següent manera:

```
$data = $html->find('table.tablaTiras', 0)->children(1)->find('tr');
```

S'insereixen les parades que es troben a les rutes, ja que algunes no apareixen a la llista de totes les parades, amb un `REPLACE INTO`. Les rutes pròpiament dites s'insereixen a la corresponent taula de base de dades.

Per obtenir la ruta de tornada, cal fer una excepció que en la resta de consultes no cal. La pàgina de l'AMB està implementada en ASP i en el cas de les rutes, no permet enviar els paràmetres del combo de rutes per GET. Cal enviar-lo per POST i a més, enviar també l'estat de la vista i de les validacions del formulari. Això ho farem de la següent manera:

- Configurem la petició HTTP a realitzar:

```
$request = array(
    'http' => array(
        'method' => 'POST',
        'content' => http_build_query(array(
            '__VIEWSTATE' => $viewState,
            '__EVENTVALIDATION' => $eventValidation,
            'ddlViajes' => $ddlViajes,
            'btVisualizarViaje' => 'anything'
        )),
        'timeout' => 300,
        'header' => "Content-type: application/x-www-form-urlencoded"
    )
);
```

- Creem el context:

```
$context = stream_context_create($request);
```

- I en obtenir la URL passem el context com a paràmetre:

```
$html = file_get_html($url, FALSE, $context)
```

La ruta s'obté de la mateixa manera que la d'anada.

Si hi ha més de dues rutes, s'obtenen de manera igual que la de tornada. En canvi, si només n'hi ha una, s'evita fer la consulta de tornada (la línia es considera circular en aquest cas).

5. A continuació obté totes les parades de la següent URL:



```
http://www.ambmobilitat.cat/Principales/ImprimirParadas.aspx?
cdp=&cpr=&esq=&mun=-1&com=-1&cab=1
```

Les dades s'obtenen de la taula de la pàgina, així:

```
$data = $html->find('table#Table1', 0)->children(1)->find('tr');
```

De cada línia, se separen els elements del mobiliari (separats per comes) i s'insereixen un a un mitjançant **REPLACE INTO**, inserint també les relacions amb la parada.

S'obtenen les coordenades latitud/longitud a partir de les coordenades UTM X/Y fent una crida a una URL de l'ICC:

- En aquest cas, per motius pràctics, ja que el format retornat pel servei és JSON, utilitzem directament CURL per accedir a la URL. Ho fem a la funció `get_lat_long_from_utm_ed50` de la següent manera:

```
$url = "http://www.icc.cat/geocalc64bits/geocalc.icc";

$fields =
"ref1=100&ref2=800&coord1=0&coord2=1&h1=-1&h2=-1&g1=-1&g2=-1&x_lon=$utm
_x&y_lat=$utm_y&z_h=0";

$useragent="Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.17 (KHTML,
like Gecko) Chrome/24.0.1312.68 Safari/537.17 PHP/5.0";

$ch = curl_init();

curl_setopt($ch, CURLOPT_USERAGENT, $useragent); //set our user agent
curl_setopt($ch, CURLOPT_POST, 11); //set how many paramaters to post
curl_setopt($ch, CURLOPT_URL,$url); //set the url we want to use
curl_setopt($ch, CURLOPT_POSTFIELDS,$fields);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, TRUE);

$result= curl_exec ($ch) or return_error(2,"Unable to download page
from ICC"); //execute and get the results

curl_close ($ch);
```

- Extraïem els valors del codi retornat:

```
$json = json_decode($result, true);
$lat = $json['conversio'][0]['yLatOut'][0];
$long = $json['conversio'][0]['xLonOut'][0];
```

- Finalment, insereix a la base de dades les diferents ciutats de les parades i si tot ha anat bé, fa *commit*.

En cas que es produeixi algun error identificat en el procés, es fa *rollback* a la base de dades, i es retorna un codi d'error, segons la codificació següent:

- Codi 0: Sense errors o no s'ha detectat l'error.
- Codi 1: Error d'SQL.
- Codi 2: Error contactant amb el web de l'ICC.
- Codi 3: Error contactant amb el web de l'AMB.

Dins del procés hi ha un conjunt de funcions que es criden per obtenir el nom de les ciutats,

elements de mobiliari i parades, de manera que s'emmagatzemin totes amb un estil concret (primera lletra majúscula i la resta en minúscules, sense espais entre les cometes, i amb espais als guions que indiquen cantonades). No les creiem suficientment rellevants com per indicar-les aquí, però les podeu consultar al codi.

#### 4.1.2. Processos “cities\_list”, “companies\_list” i “lines\_list”

Aquests processos llegeixen de les corresponents taules de la base de dades, i retornen un text on es mostra el nom dels elements separats cadascun per un salt de línia.

#### 4.1.3. Procés “nearest\_stop”

Aquest procés llegeix el paràmetre d'entrada GET `latlong`, que ha de ser una latitud i longitud decimal, separades per un guió.

Posteriorment llegeix totes les parades de la base de dades, i calcula la que té una distància menor a la latitud i longitud indicades.

El procés retorna únicament l'identificador de la parada més propera.

#### 4.1.4. Procés “routes\_list”

Aquest procés obté la línia i parada a consultar dels paràmetres GET `line_id` i `stop_id`, respectivament.

Calcula la ruta que cal mostrar per defecte, si és possible (vegeu l'apartat “Límits”), i retorna totes les rutes amb les seves parades, amb el següent format:

```
origen_ruta_per_defecte-destinació_ruta_per_defecte
id_origen_ruta1|nom_origen_ruta1|id_destinació_ruta1|nom_destinació_ruta1|id_parada1:nom_parada1; [...] ;id_paradaN:nom_paradaN
[...]
id_origen_rutaN|nom_origen_rutaN|id_destinació_rutaN|nom_destinació_rutaN|id_parada1:nom_parada1; [...] ;id_paradaN:nom_paradaN
```

En cas que no hi hagi ruta per defecte tindrà aquesta cadena en el seu lloc: **NONE**.

#### 4.1.5. Procés “stops\_list”

Aquest procés llegeix la taula de parades de la base de dades, i retorna un text amb totes les parades, amb una línia per parada amb el següent format:

```
id_parada;nom_parada;latitud_parada;longitud_parada;línies_que_paren_a_la_parada
```

Les línies de la parada estan separades per comes.

#### 4.1.6. Procés “search\_stop”

Aquest procés pren els paràmetres GET `name`, `company`, `line` i `city` i retorna les primeres 100 parades de la base de dades que coincideixen amb els criteris indicats, amb el següent format per a cada parada, separades per salts de línia:

```
id_parada;nom_parada
```

En cas que no hi hagi resultats, es retorna la cadena **NO RESULTS**.

#### 4.1.7. Procés “stop\_info”

Aquest procés és l'encarregat d'obtenir les dades de properes arribades en temps real. El procés següent és el següent:

1. Obté l'identificador de la parada a consultar a partir del paràmetre GET **stop\_id**.
2. Obté el resultat de la URL següent:

**http://www.ambmobilitat.cat/AMBtempsbus**

Ho aconsegueix passant-li el paràmetre per POST configurant així la petició:

```
$request = array(
    'http' => array(
        'method' => 'POST',
        'content' => http_build_query(array(
            'codi' => $code
        )),
        'timeout' => 60,
        'header' => "Content-type: application/x-www-form-urlencoded"
    )
);
```

3. Obté els elements de la llista retornada per la pàgina, amb la següent consulta:

```
$data = $html->find('li b');
```

Es tracta el text obtingut per extreure'n l'identificador de la línia i el temps restant.

4. Obté la llista de línies que passen per la parada de la base de dades, i afegeix a la llista de temps obtinguts les que no hi són, amb temps -1.
5. Si no hi ha cap línia que passi per la parada, i la parada no té nom, es considera que no existeix i es retorna l'error corresponent.
6. S'obté el tipus de parada (marquesina, pal o pantalla solar) des de la base de dades, per tal de retornar-ho a l'aplicació.
7. Es retornen les dades en el següent format en cas que el resultat sigui correcte:

```
status:ok
stop_id:[identificador de la parada]
stop_name:[nom de la parada, pot estar en blanc]
stop_type:[tipus de parada: "stick", "shelter" o "solar"]
lines_and_times: L1|0|Destinació L1;L2|3|Destinació L2;L4|5|Destinació L4;
[...];LN|-1|Destinació LN
```

En cas de detectar un error, es retorna aquest resultat:

```
status:ko
error:[codi d'error: 1=error fatal, 3=error en obtenir pàgina de l'AMB,
4=parada no trobada]
```

```
cause:[causa detallada de l'error]
```

## 4.2. Implementació del refresc de les dades des de l'aplicació

El refresc de les dades es realitza en fer clic al botó “Refresca” de `StationFragment`. Un cop iniciat, es defineix el booleà `isRefreshing` a cert, es crida `supportInvalidateOptionsMenu()` perquè es tornin a pintar les icones de la barra d'accions i es llança la tasca asíncrona “`DownloadStationTask`”. Una vegada la tasca acaba, s'executa el mètode corresponent per mostrar els resultats o els possibles errors produïts, es defineix `isRefreshing` a fals, i es torna a cridar `supportInvalidateOptionsMenu()` perquè torni a pintar les icones de la barra d'accions.

Al mètode `createOptionsMenu()`, que és cridat pel sistema quan es necessita pintar les icones de la barra d'accions, hi afegim la condició que el botó “Refresca” el dibuixi únicament quan `isRefreshing` sigui fals, i hi dibuixi una animació de progrés en cas contrari.

D'aquesta manera, no es pot començar un refresc addicional fins que no ha acabat el primer, i es mostra gràficament que s'estan actualitzant les dades.

## 4.3. Implementació de la interfície en mode mapa

El fragment del mapa està implementat a partir d'una subclasse de `SupportMapFragment`, que és la classe proporcionada per la biblioteca Google Play Services.

Dins d'aquesta subclasse, hi hem afegit la lògica necessària als mètodes `onCreate`, `onResume`, `onCreateView`, etc.

Dins d'aquesta lògica especial hi incloem pintar-hi els marcadors amb les ubicacions de les parades. Carreguem totes les parades a partir d'una URL, i utilitzem un codi per pintar únicament les visibles a la zona del mapa que tenim ampliada. Per tal d'evitar la lentitud en alguns dispositius en mostrar molts marcadors, si l'ampliació no és prou gran no es mostra cap parada i es mostra un missatge informatiu “Amplia més per veure les parades”.

En cas que Google Play Services no estigui disponible al dispositiu (ho comprovem amb el mètode `GooglePlayServicesUtil.isGooglePlayServicesAvailable()` a `SearchFragment`), no es mostra l'opció de consultar el mapa, ja que donaria un error en no disposar de la biblioteca necessària.

## 4.4. Implementació de l'escaneig per QR

Pel que fa a la implementació de l'escaneig per QR, podem distingir-hi dues casuístiques diferents: accés intern o extern.

### 4.4.1. Obertura de QR interna

L'obertura de QR es fa internament prement el botó “Escaneja un QR” de la pantalla de cerca (`SearchFragment`). Aquest botó utilitza la classe `IntentIntegrator` de ZXing per llançar un intent a l'aplicació “Barcode Scanner”, la qual obre la pantalla per llegir un QR amb la càmera, i en llegir-lo retorna el resultat a través de `onActivityResult()`. Allà s'extreu el codi de la URL proporcionada i es crida la pantalla `StationActivity/StationFragment`.

En cas que l'aplicació Barcode Scanner no estigui instal·lada, en fer clic al botó se n'informa a l'usuari i se li permet instal·lar-la des de Google Play.

#### 4.4.2. Obertura de QR externa

Aquest sistema permet que l'usuari utilitzi el seu lector de QR preferit i escanegi directament el QR sense obrir l'aplicació. Si l'URL coincideix amb els patrons de les URLs de TMB o de l'AMB, se li demanarà si vol obrir l'URL amb “Proper bus Barcelona”. Si l'usuari marca la casella “Sempre” a partir d'aleshores qualsevol URL que coincideixi amb aquell patró s'obrirà amb l'aplicació.

Això ho fem definint els següent paràmetres a l'AndroidManifest.xml:

```
<activity
    android:name="cat.ereza.properbusbcn.MainActivity"
    android:label="@string/app_name" >
    <intent-filter>
        <action android:name="android.intent.action.MAIN" />
        <category android:name="android.intent.category.LAUNCHER" />
    </intent-filter>
    <intent-filter>
        <action android:name="android.intent.action.VIEW" />
        <category android:name="android.intent.category.DEFAULT" />
        <category android:name="android.intent.category.BROWSABLE" />
        <data android:scheme="http"
            android:host="www.ambmobilitat.cat"
            android:pathPrefix="/ambtempsbus" />
        <data android:scheme="http"
            android:host="www.AMBmobilitat.cat"
            android:pathPrefix="/AMBtempsbus" />
        <data android:scheme="http"
            android:host="www.tmb.cat"
            android:pathPrefix="/mobile/pwtmb" />
    </intent-filter>
</activity>
```

#### 4.4.3. Extracció del codi des de les URLs

El codi s'extreu de les URLs mitjançant el mètode `extractStopFromQrUrl` de la classe `BusHelper`, que està implementat d'aquesta manera:

```
public static Integer extractStopFromQrUrl(String contents) {
    if (contents.contains(Globals.QR_BEGINNING_AMB_UPPER)) {
        contents = contents.replace(Globals.QR_BEGINNING_AMB_UPPER, "");
    }
    else if (contents.contains(Globals.QR_BEGINNING_AMB_LOWER)) {
        contents = contents.replace(Globals.QR_BEGINNING_AMB_LOWER, "");
    }
}
```

```

else if (contents.contains(Globals.QR_BEGINNING_TMB)) {
    contents = contents.replace(Globals.QR_BEGINNING_TMB, "");
}
else if (contents.contains(Globals.QR_BEGINNING_TMB_MQ)) {
    contents = contents.replace(Globals.QR_BEGINNING_TMB_MQ, "");
}
try{
    return Integer.parseInt(contents);
}
catch (NumberFormatException e){
    return null;
}
}

```

Les constants són les següents:

```

public static final String QR_BEGINNING_AMB_UPPER =
    "http://www.AMBmobilitat.cat/AMBtempsbus?codi=";
public static final String QR_BEGINNING_AMB_LOWER =
    "http://www.ambmobilitat.cat/ambtempsbus?codi=";
public static final String QR_BEGINNING_TMB =
    "http://www.tmb.cat/mobile/pwtmb?action=menu&codi_parada=";
public static final String QR_BEGINNING_TMB_MQ =
    "MQ";

```

Les dues primeres es troben al web i parades de l'AMB, i les dues restants a les parades de TMB (la segona és un codi de barres petit que hi ha a la marquesina).

## 4.5. Implementació de la coloració de les línies

Donat que al web de l'AMB no es distingeix el color de les línies més enllà de vermell (TMB) o groc (resta), i que no és possible extreure'l mitjançant *web scraping* (es tracta d'imatges), hem decidit implementar la coloració dins de l'aplicació. Aquests colors són els que es mostren a la llista de properes arribades, i representen el color de la línia. La implementació resideix al mètode `formatViewAccordingToLine()` de la classe `BusHelper`, i és la següent:

```

public static void formatViewAccordingToLine(TextView textView, Resources
resources) {
    String line = textView.getText().toString();

    try{
        Integer.valueOf(line);
        //if doesn't throw exception: it's a number
        //TMB LINE
        textView.setTextColor(resources.getColor(R.color.tmb_normal_fg));
    }
}

```

```

        textView.setBackgroundColor(resources.getColor(R.color.tmb_normal_bg));
    }
    catch (NumberFormatException e) {
        //NON-TMB LINE
        if (line.startsWith("N")) {
            //NIT BUS
            textView.setTextColor(resources.getColor(R.color.nit_fg));
            textView.setBackgroundColor(resources.getColor(R.color.nit_bg));
        }
        else if (line.startsWith("V3") || line.startsWith("V7") ||
                line.startsWith("V17") || line.startsWith("V21")) {
            //TMB V LINE
            textView.setTextColor(resources.getColor(R.color.tmb_v_fg));
            textView.setBackgroundColor(resources.getColor(R.color.tmb_v_bg));
        }
        else if (line.startsWith("D20")) {
            //TMB D LINE
            textView.setTextColor(resources.getColor(R.color.tmb_d_fg));
            textView.setBackgroundColor(resources.getColor(R.color.tmb_d_bg));
        }
        else if (line.startsWith("H6") || line.startsWith("H8") ||
                line.startsWith("H10") || line.startsWith("H12") ||
                line.startsWith("H16")) {
            //TMB H LINE
            textView.setTextColor(resources.getColor(R.color.tmb_h_fg));
            textView.setBackgroundColor(resources.getColor(R.color.tmb_h_bg));
        }
        else {
            //AMB LINE
            textView.setTextColor(resources.getColor(R.color.amb_fg));
            textView.setBackgroundColor(resources.getColor(R.color.amb_bg));
        }
    }
}

```

Aquest mètode és cridat en pintar la llista des del mètode `getView` de `LinesAndTimesAdapter`, i també en pintar la finestra d'informació d'una parada al mapa.

## 4.6. Dificultats trobades en la implementació

A continuació exposem les principals dificultats trobades durant el desenvolupament de l'aplicació,

així com la solució per la qual hem optat.

### 4.6.1. Canvis en la web de l'AMB

L'obtenció de dades mitjançant *web scraping* no és especialment complexa, però sí que pot resultar complicada quan es produeix un canvi en el codi de la pàgina origen de les dades. Sovint cal adaptar no només el selector de les dades sinó també altres punts del codi.

Des que vam iniciar el projecte, el febrer del 2013, s'han produït tres canvis de format, que han estat els següents:

1. Canvi del nom del camp codi del formulari. Es tracta d'un canvi senzill ja que únicament va ser necessari canviar el paràmetre POST `codi_form` per `codi`, però l'aplicació va estar unes hores retornant un error.
2. Incoherència de dades. Una de les parades de bus apareixia duplicada a la taula de totes les parades. Va ser necessari modificar l'algorisme per evitar duplicitats.
3. Canvi del format de les dades des d'una taula a una llista. Va ser necessari adaptar el selector de les dades, i modificar l'algorisme que obté la llista de línies, ja que en cas d'error es mostra també una línia d'error que es podia confondre amb dades.

És difícil preveure com poden afectar possibles canvis de format de la pàgina, de manera que la solució cal trobar-la una vegada s'han produït. Per minimitzar l'interval en què l'aplicació no funciona, hem construït un *script* que realitza una petició cada hora, i si el codi de la pàgina ha canviat ho notifica per correu a l'administrador. Aquest script el trobareu al fitxer `page_checker.sh`.

### 4.6.2. “Bugs” de les biblioteques

En fer ús de diverses biblioteques de tercers, ens exposem sempre als errors de programació que puguin tenir. A continuació detallem els que hem trobat i la seva solució.

1. Barra d'acció congelada en Android <4.0: A l'ActionBarSherlock 4.2 hi havia un error[16] que feia que, en cas de no tenir cap element del menú a l'activitat però sí al fragment mostrat, el menú quedava congelat una vegada es premia el botó. La solució va ser crear un menú fictici la primera vegada que es construïa el menú, i eliminar-lo en futures iteracions. D'aquesta manera:

```
public boolean onCreateOptionsMenu(Menu menu) {
    if (firstTimeMenu) {
        menu.add("").setEnabled(false);
        firstTimeMenu = false;
    }
    return super.onCreateOptionsMenu(menu);
}
```

Aquest codi ja no és necessari en l'última versió 4.3.

2. Error “Can not perform this action after onSaveInstanceState” en retornar de la crida a `IntentIntegrator` per llegir un QR[17]: la biblioteca de suport per utilitzar fragments en versions anterior a la 3.0 té un error que fa que `onActivityResult()` s'executi abans que



`onStart()` [18]. La solució ha estat crear una variable temporal a `onActivityResult()` on guardar el resultat de l'execució, i tractar-lo a l'`onResume()`.

3. La IU no s'actualitza en fer ús del `onItemSelectedListener` d'un *spinner* amb HoloEverywhere[19]: la solució ha estat endarrerir-ne l'execució llançant els canvis en un fil independent:

```
Handler han = new Handler();
han.postDelayed(new Runnable() {
    @Override
    public void run() {
        getActivity().runOnUiThread(new Runnable() {
            @Override
            public void run() {
                updateSelectedRoute();
            }
        });
    }
}, 0);
```

4. Els diàlegs es tanquen en canviar l'orientació: la biblioteca de compatibilitat de fragments per a Android <3.0 té un error que fa que, si està definit `setRetainInstance(true)`, en rotar el dispositiu els diàlegs (`DialogFragment`) es tanquen[20]. La solució és afegir el següent codi:

```
@Override
public void onDestroyView(){
    Dialog dialog = getDialog();
    // Work around bug
    if ((dialog != null) && getRetainInstance()){
        dialog.dismiss();
    }
    super.onDestroyView();
}
```

5. Els diàlegs no mostren el títol en Android <4.0: error de la biblioteca HoloEverywhere, que fa que els diàlegs no mostrin el títol. La solució és simular que són `AlertDialog` en lloc de `Dialog`:

```
@Override
public Dialog onCreateDialog(Bundle savedInstanceState) {
    setDialogType(DialogType.AlertDialog); //To workaround bug
    return super.onCreateDialog(savedInstanceState);
}
```

Tots aquests errors estan correctament comentats al codi amb les seves URLs de referència perquè

sigui senzill identificar-los en un futur.

### 4.6.3. Problema amb la publicitat en un dispositiu

Hi ha un dispositiu concret que dona errors en mostrar la publicitat d'AdMob. Es tracta de l'LG Optimus L3.

La solució ha estat amagar els anuncis en aquest model concret, creant un mètode que retorna cert si el dispositiu té problemes per mostrar els anuncis. Si és així, no es mostren. El codi és aquest:

```
public static boolean hasProblemsWithAds() {  
    return ("LG-E400".equals(Build.MODEL) &&  
        "e0_vdf_eur".equals(Build.PRODUCT)) ;  
}
```

## 4.7. Límits de la implementació

Els límits de la implementació realitzada vénen donats principalment per les dades proporcionades i la manera d'obtenció de les mateixes. A continuació els exposem.

### 4.7.1. Durada de l'execució i límits d'ample de banda

El fet d'obtenir les dades des dels servidors de l'AMB implica que explotem el seu ample de banda. Si bé un ús normal segurament no tingui implicacions, en cas de realitzar moltes peticions podríem trobar-nos amb el fet que ens deneguin l'accés al seu servidor. És per això que hem d'intentar fer un ús racional dels recursos.

Actualment, s'accedeix de forma massiva al servidor una vegada al dia, a les 4:00 de la matinada, hora en què creiem que l'afectació per als servidors serà mínima ja que hi ha pocs usuaris. No obstant això, el procés d'obtenció de dades dura aproximadament uns 40 minuts, principalment pel fet d'haver de consultar les rutes de cada línia.

Pel que fa a les consultes en temps real, la mitjana al llarg del dia és d'unes 7 consultes per minut, amb moments puntuals d'aproximadament unes 20 consultes per minut.

### 4.7.2. Direcció de les línies de bus

Les dades en temps real no permeten obtenir en quina direcció es mou el bus. Per tant, per a les parades on només s'atura el bus en una direcció, podem mostrar-ne la direcció ja que la podem deduir de la ruta.

No obstant, per a les parades on el bus para en més d'una direcció, i on seria més útil aquesta informació, no podem determinar-la per manca de dades. En aquests casos, la destinació de la línia apareix en blanc.

Aquest problema no és únic de la nostra aplicació, ja que l'aplicació oficial AMBTempsbus té el mateix problema. No obstant, a les pantalles informatives de les parades sí que es disposa de la direcció dels busos.

## 4.8. Possibles millores

Exposem aquí algunes de les possibles millores pel que fa a la implementació del projecte.

### 4.8.1. Evitar múltiples crides al servei de l'ICC

Actualment, dins del procés “fetch\_all\_data” es crida un servei web de l'ICC per transformar les coordenades UTM a latitud i longitud. Aquesta crida es fa una vegada per a cada parada, la qual cosa implica que fem quasi 5000 peticions al web de l'ICC. Seria recomanable modificar-la perquè fes crides de 30 en 30, que és el màxim que permet fer el servei web. D'aquesta manera disminuiríem l'ús d'ample de banda.

### 4.8.2. Comentaris al codi

Si bé el codi relatiu als processos d'obtenció de dades (la part més complexa del projecte) està degudament comentat, seria interessant millorar els comentaris al codi de l'aplicació Android, que, llevat d'alguns comentaris puntuals en mètodes de la classe `BusHelper`, no existeixen.

### 4.8.3. Ús de “caché” al mapa

Actualment, cada vegada que s'obre el fragment del mapa, es carrega la llista sencera de parades de bus des del servidor. Això implica que cada vegada que s'obre, es descarreguen aproximadament 150 KB. Tot i que no sol impactar excessivament en la rapidesa de càrrega, sí que és important intentar reduir l'ús de dades de cara als usuaris que tenen limitacions d'ample de banda, ja que fer un ús continuat de l'aplicació pot resultar en un ús més elevat d'ample de banda.

Una possible solució seria desar la llista en una *cache* juntament amb la data d'obtenció, i refrescar-la només si fa més d'un dia que està emmagatzemada (les parades només s'actualitzen una vegada al dia).

## 5. Temporalització i cost de desenvolupament

A continuació tractarem tot allò relatiu al cost i temporalització del desenvolupament realitzat.

### 5.1. Cost de desenvolupament

Hem elaborat la següent taula per realitzar un càlcul del temps i cost necessari per realitzar el projecte:

CONCEPTE	HORES DESENV.
<b>Aplicació Android</b>	
Crear estructura bàsica del projecte	2
Creació de recursos gràfics (icones)	4
Creació pantalla inicial	4
Creació pantalla de la parada	10
Creació pantalla de cerca	6
Impl. obtenció dades temps real (aplicació)	8
Impl. colors línies	1
Impl. obtenció cerca i resultats	6
Impl. pantalla quant a	2
Adaptació a tablets	8
Impl. favorites i últimes parades	6
Impl. obtenció rutes	3
Creació pantalla del mapa	6
Consulta per QR	4
Impl. parada més propera	2
Creació pantalla de rutes	3
<b>Processos</b>	
Disseny i creació BBDD	4
Obtenció de dades en temps real	8
Obtenció i tractament de dades diari	12
Processos de retorn de dades a partir de BBDD	6
Scripts de suport (comparació, etc.)	2
<b>MULTIPLICADOR (anàlisi, proves, documentació, etc...)</b>	<b>15,00%</b>
<b>TOTAL</b>	<b>124 h</b>
<b>TARIFA</b>	<b>30,00 €/h</b>
<b>IMPORT</b>	<b>3.720,00 €</b>
AMB IVA	4.501,20 €

Hem detallat les hores de desenvolupament necessàries per a cada element de l'aplicació, i hi hem afegit un 15% addicional per a proves, anàlisi i documentació del codi.

Per tant, hem obtingut que el temps estimat de desenvolupament seria de 124 hores, és a dir, unes 3 setmanes amb un programador a temps complet. L'import, considerant una tarifa de 30€/h, seria de

3.720,00€ + IVA.

## 5.2. Monetització: opcions i elecció

En plataformes mòbils, per a aplicacions desenvolupades per un mateix com és el cas d'aquest projecte, hi sol haver tres opcions de monetització, que són les següents:

- Aplicació gratuïta, mostrant publicitat: Es tracta de l'opció més utilitzada en aplicacions Android per a aplicacions senzilles, ja que sovint els usuaris no volen pagar per les aplicacions. Sol generar uns ingressos variables en funció dels anuncis mostrats, clics fets pels usuaris i país d'origen dels usuaris.
- Aplicació de pagament: Consisteix en demanar un pagament a l'usuari abans de poder descarregar l'aplicació. Una vegada comprada i descarregada, l'usuari disposa de 15 minuts per retornar-la si no li agrada[21]. L'import el fixa el desenvolupador, que comparteix un percentatge dels ingressos amb la plataforma on publica l'aplicació (en aquest cas, Google Play).
- Aplicació gratuïta amb opcions extra de pagament: L'aplicació és gratuïta per a l'usuari, però pot adquirir elements extra fent pagaments dins de l'aplicació (*"in-app purchases"*). És utilitzada sovint pels jocs *"free to play"*, que en fan ús per facilitar el joc al jugador o eviten que l'usuari hagi d'estar molta estona jugant per aconseguir un objectiu, permetent comprar-lo directament. També s'utilitza en algunes aplicacions on es mostra publicitat per eliminar-la permanentment.

Per al nostre projecte, donat que totes les aplicacions amb objectius similars són gratuïtes, hem optat per la via de fer l'aplicació gratuïta mostrant publicitat.

Actualment, amb l'aplicació publicada i unes 10.000 consultes diàries, els ingressos mitjans són d'aproximadament 70€/mes, tot i que segurament vagi en augment si el nombre d'usuaris augmenta.

## 5.3. Cost d'actualització

En aplicacions mòbils, és molt important estar sempre a l'última i mantenir l'aplicació actualitzada a les funcionalitats que permet el sistema. És important també tenir en compte els possibles suggeriments dels usuaris finals de cara a implementar noves funcionalitats.

En aquest projecte, a més a més, cal tenir certa disponibilitat per realitzar canvis sobtats, degut als problemes que pot comportar un canvi en el format d'origen de les dades.

És per això que cal tenir en compte que el cost de l'aplicació no és únicament el cost inicial de desenvolupament, sinó que hi ha un cert cost d'actualització que pot variar al llarg del temps i que, depenent del projecte, pot ser difícil d'estimar.

## 6. Conclusions i possibles extensions

Finalment, en aquest punt exposem les conclusions a què hem arribat durant la realització del projecte, així com possibles extensions futures al mateix, que no han estat possibles per motius diversos.

### 6.1. Conclusions

A la vista dels resultats del projecte, sembla que val la pena intentar esforçar-se per aconseguir una aplicació amb una millor usabilitat encara que partim d'un punt en què hi ha diverses aplicacions competidores. Hem pogut superar els obstacles i aconseguir una aplicació que fa tot el que fan les seves competidores juntes, i fins i tot més coses.

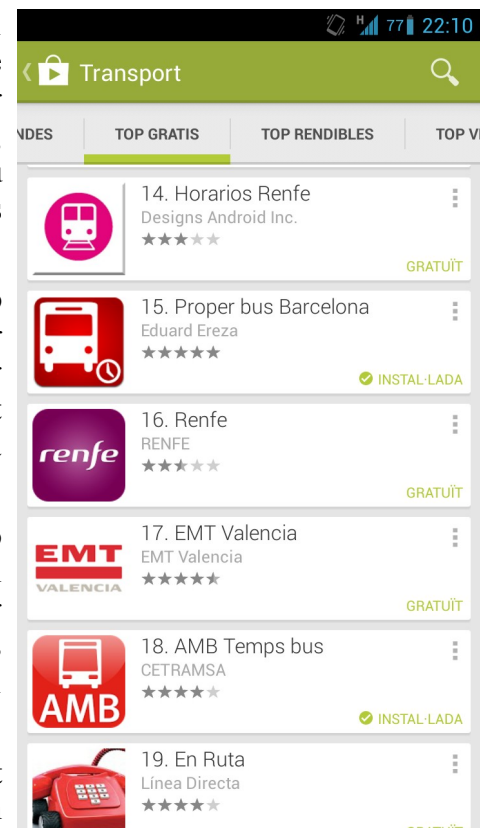
En el camí, tot i que ja havíem treballat amb Android, hem après a utilitzar la biblioteca HoloEverywhere (i també que és bastant inestable pel que fa a *bugs*) per disposar de la mateixa interfície arreu, hem aprofundit en el coneixement dels fragments i ens hem introduït de ple en el món del *web scraping*.

Pel que fa a l'aplicació, publicada al Google Play i descarregable per qui ho desitgi, hem aconseguit més de 20.000 descàrregues en 5 mesos, i fins i tot hem aconseguit ser la 15a aplicació de transport més descarregada a Espanya, superant l'aplicació oficial de la qual obtenim les dades (vegeu la imatge del costat). La puntuació que ens atorguen els usuaris, amb més de 100 valoracions, és de 4,8 sobre 5.

Probablement si haguéssim de viure d'aquesta aplicació no l'hauríem desenvolupat, ja que els ingressos obtinguts per publicitat no compensen el temps que ha calgut (i cal) invertir en el projecte, almenys per ara, però sí que ens ha estat molt útil per aprofundir en el coneixement d'Android i donar-nos a conèixer.

Personalment, no esperàvem ni molt menys una repercussió com la que ha tingut l'aplicació, ni els comentaris que ens han atorgat els usuaris. Partíem d'una necessitat senzilla, consultar el temps d'espera per al proper bus, i fer-ho de manera que fos senzill per als usuaris. Sembla que l'esforç ha valgut la pena i els usuaris n'han quedat satisfets.

No val abaixar la guàrdia, però, i caldrà seguir actualitzant l'aplicació tan bon punt apareguin noves dades que puguem emprar per millorar-la, o bé noves idees.



### 6.2. Extensions futures

De cara al futur, és possible expandir aquest projecte en diversos aspectes.

En primer lloc, es pot adaptar l'aplicació a altres ciutats. Es pot crear un "Proper bus Lleida", un "Proper bus Madrid" o un "Proper bus Londres", tenint en compte les característiques de la xarxa de bus i les dades disponibles en cada cas.

Pel que fa a ampliar l'aplicació existent, a part de les millores d'implementació que ja hem comentat en anteriors apartats, es poden implementar diverses funcionalitats que de ben segur serien útils per als usuaris: mostrar els horaris de les línies, reordenar els preferits, mostrar la ruta des d'on és l'usuari fins a una destinació concreta incloent-hi canvis de línia, mostrar les incidències programades de les línies, etc.

També podria resultar interessant, donat el relatiu èxit de la versió per a Android, portar l'aplicació a altres plataformes com ara iOS, BlackBerry o Windows Phone.

## Apèndix 1. Instal·lació del servidor de dades

Per realitzar la instal·lació de tot el programari necessari del servidor, partirem de la instal·lació d'una distribució Ubuntu 12.04 LTS, versió servidor. No detallarem aquí tots els passos necessaris per instal·lar el sistema operatiu, però sí tots els passos a seguir una vegada instal·lat el sistema bàsic.

Una vegada tinguem el sistema operatiu instal·lat, instal·larem els paquets necessaris amb aquesta ordre:

```
sudo apt-get install apache2 php5 php5-cli libapache2-mod-php5 mysql-server-5.5  
php5-mysql php5-curl sendmail diffutils
```

L'instal·lador ens preguntarà una contrasenya per a l'usuari root de MySQL, hi introduïrem la contrasenya desitjada.

A continuació crearem l'usuari de la nostra base de dades i l'estructura bàsica.

Copiarem el fitxer `proper_bus_bcn_mysql.sql` al directori on treballem, i executarem la següent ordre:

```
mysql -uroot -p
```

Hi introduïrem la contrasenya definida anteriorment. A continuació, crearem la base de dades i l'usuari, com s'indica:

```
CREATE USER 'proper_bus_bcn'@'localhost' IDENTIFIED BY 'contrasenya_desitjada';  
GRANT ALL PRIVILEGES ON proper_bus_bcn.* TO 'proper_bus_bcn'@'localhost';  
FLUSH PRIVILEGES;  
USE proper_bus_bcn;  
\. proper_bus_bcn_mysql.sql
```

Una vegada fet això podem prémer Ctrl-D per sortir de l'interpret de MySQL, i ja tindrem la base de dades definida.

A continuació cal editar els fitxers `config.php` i definir-hi l'usuari, contrasenya i nom de la base de dades en els camps corresponents.

Per continuar, definirem el lloc web a la configuració de l'Apache. Podem definir-ho com creguem convenient, depenent de l'estructura que vulguem al nostre servidor. En aquest cas assumim que el servidor és exclusiu per al nostre projecte..

Editarem el fitxer `/etc/apache2/sites-enabled/000-default` i canviarem el `DocumentRoot` al directori on tinguem els processos emmagatzemats. Si no el canviem, per defecte és `/var/www/`.

A continuació reiniciarem l'Apache perquè agafi la nova configuració:

```
sudo service apache2 restart
```

Provarem de visitar el nostre servidor des d'un navegador, per exemple:

```
http://host_del_servidor/proper_bus_bcn/stop_info.php?stop_id=1269
```

Si rebem resposta començant per “`status:`” és que tot és correcte. En cas contrari, cal revisar la configuració.

Finalment, cal configurar les tasques programades que obtenen les dades cada nit. Ho farem mitjançant cron:



```
sudo crontab -e
```

Afegirem les següents línies al cron existent:

```
#Checks if AMBTempsBus page has changed - hourly
```

```
0 * * * * /var/www/proper_bus_bcn/checker/page_checker.sh > /dev/null
```

```
#Fetches data from AMB - v2 - daily at 4:00
```

```
0 4 * * * cd /var/www/proper_bus_bcn/fetcher/ && ./fetch_all_data.sh
```

Lògicament també hi podem configurar línies addicionals per fer còpies de seguretat o el que necessitem.

Una vegada fet això, cal editar també els fitxers `fetch_all_data.sh` i `page_checker.sh` per definir-hi els destinataris del correu electrònic i, en el primer fitxer, l'usuari i contrasenya amb què es fa l'exportació de les bases de dades.

Quan haguem completat totes aquestes tasques, el servidor estarà correctament preparat per a l'aplicació.

## Apèndix 2. Instal·lació de l'entorn de desenvolupament

Per a la instal·lació de l'entorn de desenvolupament podem optar per dues opcions: configurar-lo nosaltres o optar per un paquet d'Eclipse ja configurat per a Android, com els que ofereixen organitzacions com CatDroid[22] o, més recentment, Google[23]. En aquest projecte detallarem la configuració manual, ja que és la que hem seguit.

Inicialment, descarregarem el paquet “Eclipse Standard” des del lloc web oficial[24] i el descomprimirem al nostre directori de preferència. Ara podem obrir ja l'Eclipse.

A continuació, instal·larem l'extensió ADT[25]. Per fer-ho, anirem al menú “Help -> Install new software”, i afegirem el lloc següent:

<https://dl-ssl.google.com/android/eclipse/>

Instal·larem tots els paquets de programari que hi ha en aquest lloc, i reiniciarem l'Eclipse si ens ho demana.

Una vegada iniciat de nou, ens demanarà descarregar l'SDK d'Android. Seguirem les instruccions de l'auxiliar i una vegada instal·lat, ja podrem treballar amb projectes Android.

Quan l'entorn ja estigui preparat, importarem els projectes que necessitem. Ho farem mitjançant el menú “File->Import...” i escollirem “Existing Projects into Workspace”. Ho farem per a tots els següents projectes: “ActionBarSherlock-4.3.0”, “GooglePlayServices-7”, “HoloEverywhere”, “HoloEverywhere-Preferences” i finalment, “ProperBusBCN”.

A continuació caldrà que definim les biblioteques de cada projecte. Per fer-ho, farem clic amb el botó dret sobre “HoloEverywhere” i escollirem “Properties”, i després “Android”. Esborrarem les dependències que hi hagi i afegirem “ActionBarSherlock-4.3.0”.

Farem el mateix a “HoloEverywhere-Preferences”, però en aquest cas afegint-hi “HoloEverywhere”.

Finalment, farem el mateix a “ProperBusBCN”, però afegint-hi tres biblioteques: “HoloEverywhere”, “HoloEverywhere-Preferences” i “GooglePlayServices-7”.

Una vegada fet això, podem fer “Project -> Clean -> Clean all projects”, i tots els projectes haurien de compilar correctament.

Ara ja tenim l'entorn i podem executar l'aplicació fent-hi clic amb el botó dret i prement “Run As -> Android Application”. Si tenim un dispositiu Android connectat a l'ordinador i amb el mode de desenvolupament activat, s'hi instal·larà i executarà automàticament.

Abans de fer-ho, però, cal que configurem algunes constants específiques del projecte, com són:

- **Les URLs de la classe `Globals`:** cal canviar-les per les del nostre propi servidor.
- **La constant `EMAIL_SUGGESTIONS`:** contindrà el correu on s'envien els suggeriments.
- **La constant `ADMOB_KEY`:** contindrà la clau per mostrar els anuncis d'AdMob (cal registrar-se a AdMob i crear un projecte per obtenir-ne la clau). Es pot executar l'aplicació sense la clau, però no es mostraran anuncis.
- **El paràmetre `ga_trackingId` del fitxer `analytics.xml`:** contindrà l'identificador del lloc de Google Analytics on s'informarà de les accions que emprèn l'usuari (cal registrar-se a Google Analytics). Es pot executar l'aplicació sense el codi, però no funcionaran les

estadístiques.

- **L' anotació de la classe `ProperBusApplication`:** ha de contenir la URL, usuari i contrasenya d'Acralyzer on informar dels errors amb ACRA (cal instal·lar un servidor on informar dels errors). Es pot eliminar l'anotació si no es disposa d'aquest sistema.
- **Configurar l'ús de Google Maps:** cal utilitzar la consola d'APIs de Google per configurar-hi l'ús de Google Maps per al conjunt d'aplicació i signatura. Ho farem seguint les instruccions de Google[26]. Caldrà introduir la clau proporcionada a l'atribut `com.google.android.maps.v2.API_KEY` de l'`AndroidManifest.xml`.

Una vegada haguem canviat tots aquests paràmetres, podrem modificar i executar l'aplicació com desitgem.

## Bibliografia

- 1: Google, "*TMB Virtual - Aplicacions d'Android a Google Play*", 2013, <https://play.google.com/store/apps/details?id=com.geomobile.tmbmobile>, consultat a 7 de juny del 2013
- 2: Google, "*Urban Step - Barcelona- Aplicacions d'Android a Google Play*", 2013, <https://play.google.com/store/apps/details?id=com.asciil64.totsmovem>, consultat a 7 de juny del 2013
- 3: Google, "*Buses Barcelona - Aplicacions d'Android a Google Play*", 2013, <https://play.google.com/store/apps/details?id=com.miguelfermin.bcnbuses>, consultat a 7 de juny del 2013
- 4: Google, "*Buzz Barcelona - Aplicacions d'Android a Google Play*", 2013, <https://play.google.com/store/apps/details?id=com.treenamics.android.buzz.free>, consultat a 7 de juny del 2013
- 5: Google, "*AMB Temps bus - Aplicacions d'Android a Google Play*", 2013, <https://play.google.com/store/apps/details?id=com.amb.ambtemps>, consultat a 7 de juny del 2013
- 6: Àrea Metropolitana de Barcelona, "*AMB - Mobilitat*", 2013, <http://www.ambmobilitat.cat/Principales/Inicio.aspx>, consultat a 7 de juny del 2013
- 7: Àrea Metropolitana de Barcelona, "*AMB Temps bus*", 2013, <http://www.ambmobilitat.cat/AMBTempsbus>, consultat a 19 de juny del 2013
- 8: Àrea Metropolitana de Barcelona, "*Avis Legal - AMB - Mobilitat*", 2013, <http://www.ambmobilitat.cat/Pie/AvisoLegal.aspx>, consultat a 27 de juny del 2013
- 9: David Ehringer, "*The Dalvik Virtual Machine Architecture*", 2010, [http://show.docjava.com/posterous/file/2012/12/10222640-The\\_Dalvik\\_Virtual\\_Machine.pdf](http://show.docjava.com/posterous/file/2012/12/10222640-The_Dalvik_Virtual_Machine.pdf), consultat a 19 de juny del 2013
- 10: Wikipedia, "*Android System Architecture*", 2013, <https://en.wikipedia.org/wiki/File:Android-System-Architecture.svg>, consultat a 11 de juny del 2013
- 11: Ed Burnette, "*How Android works: The Big Picture*", 2013, <http://www.zdnet.com/blog/burnette/how-android-works-the-big-picture/515>, consultat a 11 de juny del 2013
- 12: Google, "*Accept Developer Agreement*", 2013, <https://play.google.com/apps/publish/signup/>, consultat a 19 de juny del 2013 [requereix accés amb un compte gratuït de Google]
- 13: Google, "*Activity - Android Developers*", 2013, <http://developer.android.com/reference/android/app/Activity.html>, consultat a 19 de juny del 2013
- 14: Google, "*Fragments - Android Developers*", 2013, <http://developer.android.com/guide/components/fragments.html>, consultat a 19 de juny del 2013
- 15: Google, "*Dashboard - Android Developers*", 2013, <http://developer.android.com/about/dashboards/index.html>, consultat a 28 de juny del 2013
- 16: Autors diversos, "*After invalidateOptionsMenu() items no more clickable - Github*", 2013, <https://github.com/JakeWharton/ActionBarSherlock/issues/510>, consultat a 27 de juny del 2013
- 17: Autors diversos, "*Show DialogFragment from onActivityResult - Stack Overflow*", 2013, <http://stackoverflow.com/questions/10114324/show-dialogfragment-from-onactivityresult>, consultat a 27 de juny del 2013
- 18: Autors diversos, "*Calling DialogFragment's show() from within onActivityResult() causes IllegalStateException in ICS - Android - An Open Handset Alliance Project*", 2013, <https://code.google.com/p/android/issues/detail?id=23761>, consultat a 27 de juny del 2013
- 19: Autors diversos, "*setVisibility() is not refreshing the activity layout - Stack Overflow*", 2012,

- <http://stackoverflow.com/questions/13561436/setvisibility-is-not-refreshing-the-activity-layout>, consultat a 27 de juny del 2013
- 20: Autors diversos, "*DialogFragment dismissed on orientation change when setRetainInstance(true) is set (compatibility library)* - *Android - An Open Handset Alliance*", 2011, <http://code.google.com/p/android/issues/detail?id=17423>, consultat a 27 de juny del 2013
- 21: Google, "*Cancel purchases - Google Play Help*", 2013, <https://support.google.com/googleplay/answer/113515?hl=en>, consultat a 1 de juliol del 2013
- 22: CatDroid.org, "*Catclipse IV (Dàtils Damnificats)*", 2012, <http://catdroid.org/ca/blog/catclipse-iv-datils-damnificats-%E2%80%93-windows/>, consultat a 30 de juny del 2013
- 23: Google, "*Android SDK*", 2013, <http://developer.android.com/sdk/index.html>, consultat a 30 de juny del 2013
- 24: The Eclipse Foundation, "*Eclipse Downloads*", 2013, <http://www.eclipse.org/downloads/>, consultat a 30 de juny del 2013
- 25: Google, "*Installing the Eclipse Plugin - Android Developers*", 2013, <http://developer.android.com/sdk/installing/installing-adt.html>, consultat a 30 de juny del 2013
- 26: Google, "*Google Maps Android API v2 - Android Developers*", 2013, [https://developers.google.com/maps/documentation/android/start#the\\_google\\_maps\\_api\\_key](https://developers.google.com/maps/documentation/android/start#the_google_maps_api_key), consultat a 1 de juliol del 2013